Knowledge Portals — Ontologies at Work

[1,2]Steffen Staab and [1]Alexander Maedche
[1]Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS
mailto:{staab, maedche}@aifb.uni-karlsruhe.de
[2]Ontoprise GmbH, Haid-und-Neu Straße 7, 76131 Karlsruhe, Germany
http://www.ontoprise.de
Contact: Steffen at tel. +49-721-6084751, fax +49-721-6086580

Ca. 7200 words

Abstract

Knowledge portals provide views onto domain-specific information on the World Wide Web, thus facilitating their users to find relevant, domain-specific information. The construction of intelligent access and the provisioning of information to knowledge portals, however, remained an *ad hoc* task requiring extensive manual editing and maintenance by the knowledge portal providers. In order to diminish these efforts we use ontologies as a conceptual backbone for providing, accessing and structuring information in a comprehensive approach for building and maintaining knowledge portals. We present one research and one commercial case study that show how our approach is used in practice.

Keywords: Ontology, Knowledge Sharing, WWW

1 Introduction

"Content is king" is the traditional credo of media companies in the Web. Now that the promises of the World Wide Web in its first stage have been delivered, information pointing to other information is ubiquitous and everyone laments under the burden of finding the needle in the haystack, the real difference only comes in when these companies aim at the better provisioning of services around these contents [35]. This trend is visible in portal services like `www.yahoo.com` or `www.look-look.com`. They enlisten hoards of people to contribute information pieces that are shaped by human editors into many different facettes. Their considerations include thinking about what information pieces are there, how to structure them, who should look at them and who should provide them.

Knowledge portals in the Web are intended to help people achieve a particular task taking place in a complex setting, e.g. learning about solutions and pitfalls of Knowledge Management[1]. Hence, knowledge portal providers must act as intermediaries that structure relevant aspects of the info world for presentation on the portal in order to allow people flexible and easy access to all the contents.

At this point artificial intelligence comes in as a key enabler for helping to structure, access and provide information that has been aggregated by collaboration of people. We have developed a comprehensive approach, tools and methods, to structure, access and provide knowledge for a given do-

[1] `www.brint.com`

main in a collaborative setting on a knowledge portal[2]. The key technology for this purpose are ontologies, which provide formal means to specify a domain of interest to a group of users (cf. [15]). So far, ontologies have been used for research [1, 31] and commercial purposes (cf., e.g., [27]) of presenting and mediating information [40] in the Web, we here present a comprehensive concept for building and maintaining knowledge portals, including the part of decentral knowledge provisioning. We elaborate on the basic tools and methods, a case study serving research needs and — briefly — a commercial portal currently under development, for which our approach is used.

2 Requirements for Knowledge Portals

The aim of knowledge portals is to make knowledge accessible to users and to allow users the exchange of knowledge. Knowledge portals specialize in a certain topic in order to offer deep coverage of the domain of interest and, thus, address a *community* of users. The portals are commonly built to include community serices, such as online forums, mailing lists and news articles of relevant guises [11].[3]

Even facing an only medium size portal the amount of information that is stored becomes extremely unwieldy to present and to re-find. In particular, the common categories, like news or mailings, appear completely inadequate to deal with the information flood on their own. Hence, the question about how best to manage such a knowledge portal comes up very urgently. One reason for this is that the user will often not care so much about the document type (mailing list, magazine article, interviews), but rather about the document content, when he searches for knowledge that he needs to solve a problem or to learn about a new topic.

In fact, there exist a number of research proposals and commercial solutions that have recognized and approached this problem. For instance, MathNet[4] introduces knowledge sharing for mathematicians through a database relying on Dublin Core meta data. Altmann *et al.* [1] allow for navigating their knowledge base on Ribosomes according to an ontology, thus providing rich interlinkage and good support for the user. Further work in this direction in various guises has also been done, e.g., in [25, 13, 26], but a **comprehensive concept** for supporting the knowledge portal has been missing so far. Such overall support has to include, of course, the GUI interface for *accessing* the portal contents and thereby addressing community-specific needs, but it also needs to consider the *provisioning* of knowledge as well as the overall construction and maintenance of the portal.

2.1 Knowledge Providing

An essential feature of a knowledge portal is the easy ammendment of information in a way such that it may be easily re-found. Thereby, ammended information may come in many different legacy formats. Nevertheless, presentations of and queries for information contents must be allowed in many ways that need to be independent from the way by that information was provided originally. The knowledge portal must remain adaptable to the information sources contributed by its providers — and not vice versa.

This requirement precludes the application of database-oriented approaches (e.g., [26]), since they presume that a uniform mode of storage exists that allows for the structuring of information

[2]Cf. [11] for a comprehensive classification of portal types.

[3]Some famous knowledge portals are `www.brint.com`, `www.netacademy.org`, `www.epinion.com`, or `msdn.microsoft.com`.

[4]`http://www.math-net.de`

at a particular conceptual level, such as a relational database scheme. In the complex setting of a knowledge portal, one must neither assume that a uniform mode for information storage exists nor that only one particular conceptual level is adequate for structuring information of a particular community. In fact, even more sophisticated approaches such as XML-based techniques that separate content from layout and allow for multiple modes of presentation appear insuffcient, because their underlying transformation mechanisms (e.g., XSLT or XQL [28, 8]) are too inconvenient for integration and presentation of various formats at different conceptual levels. The reason is that they do not provide the conceptual underpinning required for proper integration of information.

In order to integrate diverse information, we require another layer besides the common distinction into document, content and layout, viz. explicit knowledge structures that may structure all the information in different formats for a community at various levels of granularity. Different information formats need to be captured and related to the common ontology:

1. Several types of metadata such as available on web pages (e.g., HTML META-tags),
2. manual provision of data to the knowledge repository, and
3. a range of different wrappers that encapsulate structured and semi-structured information sources (e.g., databases or HTML documents).

Section 6 will address these issues in detail.

2.2 Knowledge Access

Navigating through a knowledge portal that is unknown is a rather difficult task in general. Information retrieval may facilitate the finding of pieces of texts, but its usage is not suffcient in order to provide novice users with the right means for exploring unknown terrain. This turns out to be a problem particularly when the user does not know much about the domain and does not know what terms to search for. In such cases it is usually more helpful for the user to explore the portal by browsing — given that the portal is well and comprehensively structured. Simple tree-structured portals may be easy to maintain, but the chance is extremely high that an inexperienced user looking for information gets stuck in a dead-end road. Therefore, the portal must be able to present a multitude of varying views onto its contents and different ways should be possible to approach the same content. For instance, when looking for an expert in a given, but still vaguely defined domain, either one might query for research papers or one might search for projects first and then continue to have a glimpse onto corresponding expert homepages.

Here, we must face the trade-off between resources used for structuring the portal (money, manpower) and the extent to which a comprehensive navigation structure may be provided. Since information in the knowledge portal will be continually amended, richly interrelated presentation of information would usually require extensive editing, such as is done, e.g., for Yahoo. In contrast, knowledge portals should exhibit comprehensive structuring of information virtually for free.

There has been interesting research (e.g. Fröhlich et al. [14] or Kesseler [18]) that demonstrates that authoring, as well as reading and understanding of web sites, profits from conceptual models underlying document structures “in the large”, i.e. the interlinking between documents, as well as document structures “in the small”, i.e. the contents of a particular document. In addition, it shows how rich linkage in multiple directions may be constructed automatically based on the underlying conceptual structures.

Rather naturally, once a common conceptual model for the community exists and is made explicit, it is easier for the individual to access a particular site. Hence, in addition to rich interlinking between

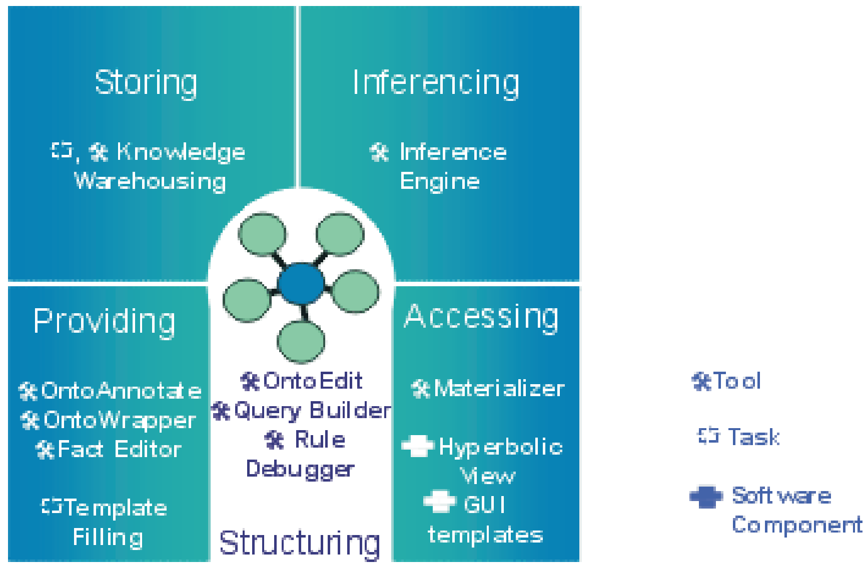

Figure 1: Knowledge Portal Architecture

document structures "in the large", comprehensive surveys and indices of contents and a large number of different views onto the contents of the portal, we require that the conceptual structure of the portal is made explicit at some point. Providing an *ontology* we meet this requirement. Section 7 will show how conceptual structures are exploited for access purposes. But before we go into the details, we provide a high-level view onto the overall architecture.

2.3 Architecture

The major division of our approach is into five modules that package different tools, tasks and software components. We have mentioned some requirements that appear at the interface sides of **accessing** and **providing** knowledge and we will elaborate on these in subsequent sections, hence we can safely ignore them here.

Knowledge Warehousing. The knowledge warehouse hosts facts, meta data about documents, and the ontology, which describes the structure of the facts and the meta data. Facts and concepts are stored in a relational database, however, they are stored in a *reified* format that treats relations and concepts as first-order objects and that is therefore very flexible with regard to changes and amendments of the ontology.

The different tasks and tools for providing knowledge feed directly into the knowledge warehouse or indirectly when they are triggered by a Web crawl. The task at this point is very similar to data warehousing, where various schemata need to be mapped to each other and views need to be maintained and integrated. In the case studies described below, we could so far restrict our attention to incoming data that was already structured according to the given ontology. Thus, the integration task has been rather negligible.

Inferencing. We exploit the inference engine, SiLRi (Simple Logic-based RDF Interpreter) described in [5]. Basically, SiLRi offers representation capabilities for RDF and F-Logic and for combinations thereof. The first is a frame-oriented representation language with an XML syntax. The second is an object-oriented logic mechanism that extends datalog with OO modelling primitives. While RDF only allows for the provisioning of facts and concept definitions, F-Logic in addition allows the querying and the use of axioms.

For our purpose, SiLRi is ideally suited because it allows for combined querying of facts and ontological concepts in one query. Hence, one may ask for questions like "show me the concept taxonomy including only those concepts for which you have some news in the last week" and, thus, dynamically adapt the portal interface. In our architecture, the knowledge warehouse is only queried via the inference engine in order to offer a uniform mode of access. However, the inference engine caches previous queries in order to deliver short response times.

Structuring. Finally, we offer several tools for structuring the portal, i.e. engineering the ontology that constitutes the background for the inference engine, and for making contents accessible. We elaborate on these tools in the section on "Structuring the Knowledge Portal", but before that we introduce two case studies as our litmus test for the validity of our approach and for illustrating some examples in the remainder of the paper.

3 Case Study: KA2Portal

The first knowledge portal that we have constructed was for the *"Knowledge Annotation Initiative of the Knowledge Acquisition community"* (KA2; *cf.* [3]). The KA2 initiative has been conceived for semantic knowledge retrieval from the web building on knowledge created in the KA community. To structure knowledge, an ontology has been built in an international collaboration of researchers. The ontology constitutes the basis to annotate WWW documents of the knowledge acquisition community in order to enable intelligent access to these documents and to infer implicit knowledge from explicitly stated facts and rules from the ontology.

Given this basic scenario, we have investigated the techniques and built the tools that we describe in the rest of this paper. Some views onto the KA2 contents may be seen in our up and running demonstration KA2 community web portal (cf. Figure 2 or `http://ka2portal.aifb.uni-karlsruhe.de`).

4 Case Study: TIME2Research-Portal

TIME is an acronym for Telecommunications, Information technology, Multimedia, and E-Business. The term "TIME market" refers to a rapidly evolving market segment with tremendous opportunities. Some of the challenges in this business segment lie in observing the market, tracking (un)successful business models, and evaluating competing or new technologies. In particular, there are many people who are not genuinely knowledgable about the TIME market and the technologies used there, but who need in-depth information such as: "*Who is selling what type of technology?*", "*Who is market leader in subsegments X?*", or "*What are peer groups of companies in sector X*".

For instance, venture capitalists are given a large number of business proposals for which they must decide quickly whether to invest in. Typically, venture capitalists are experts in financial issues of starting a company, and accordingly they use their finance expertise as a sieve to sort out the good potentials. In technical matters they would need some corresponding sieves, which they must commonly buy from a consulting company, because having the technical analyst around would turn out too expensive. From financial and technical point of views evaluation criteria of different grain sizes are used that take up different amounts of time and money. A successful proposal would run through

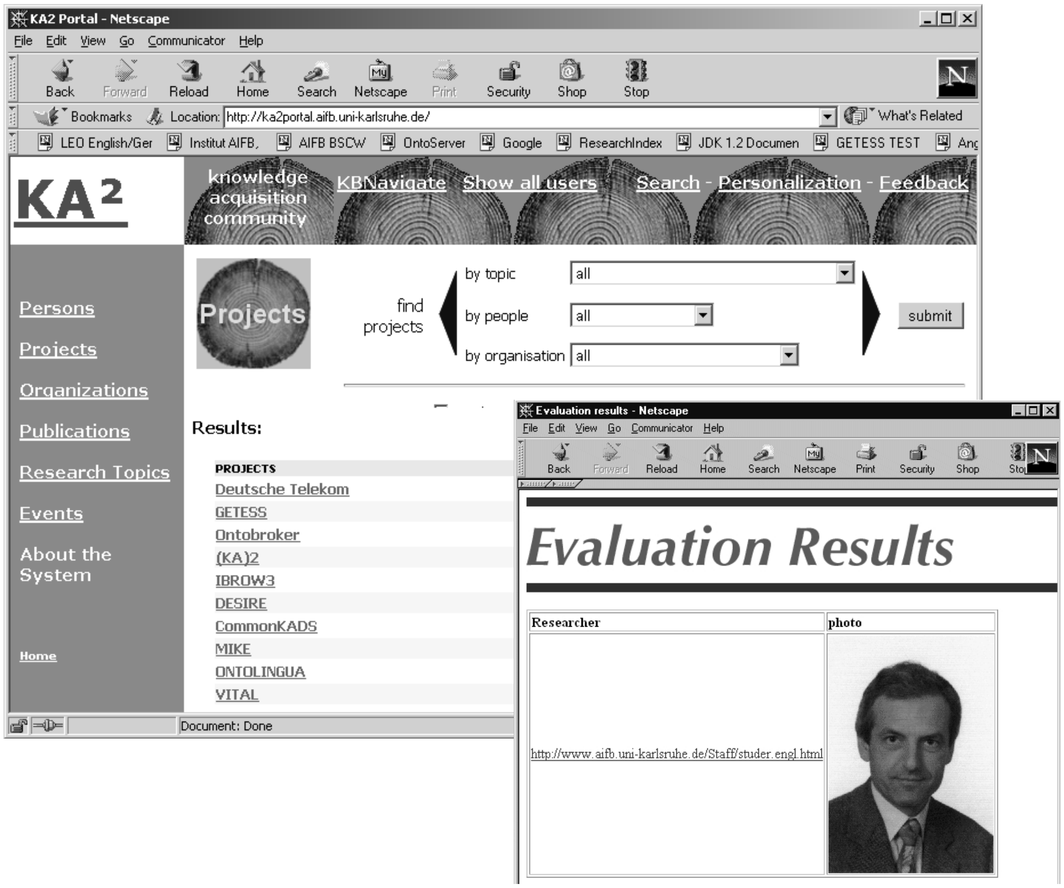

Figure 2: KA2 Portal User Interface

several evaluation cycles where increasingly fine-grained criteria and increasingly time-consuming evaluation measures are applied to sort out the good potentials from the bad.

From the outsourcing of technical expertise results a difficult problem: The duration between proposal and answer is rather long as evaluation goes through several stages. Thus, investors clog their working line and, more importantly, they may miss good chances, as proposers may turn toward other investors. Also the overall process is not very efficient, because many standard questions (such as the ones mentioned at the beginning of this section) must redundantly be researched and answered by different technical experts.

The TIME2Research knowledge portal aims at streamlining the process that the technical analysts performs, because it allows for collaborative knowledge provisioning. The portal optimizes the information delivery process between the venture capitalist and the technical expert, because it allows for decentral knowledge querying. It allows to bridge between the need for exploring the landscape and the technical expertise, because the ontology structures the relevant domain of the TIME market in terms of the one who compiles the question. The TIME2Research knowledge portal is an intriguing application, because ontologies greatly extend the capabilities of current knowledge portals in this area. Thereby, it need not solve the overall problem — evaluation in later stages will still have to be performed by technical analysts — but the venture capitalist may answer his standard questions to the portal in few minutes instead of triggering a day or week-long process.

Because the TIME2Research portal (`http://www.time2research.de/`) is still under heavy construction, we here refer the reader to a feasibility study about a system for corporate histories.[5] CHAR [2], the corporate history analyser, tackles part of what is aimed at with the TIME2Research portal. It may be seen running with few example data about past activities of M.A. Hanna at `http://char.aifb.uni-karlsruhe.de/`.[6]

5 Structuring the Knowledge Portal

Ontologies have been established for knowledge sharing and are used as a means for conceptually structuring domains of interest [40, 36]. As knowledge portals focus on particular domains, ontologies appear ideally suited to support knowledge sharing and re-use between knowledge portal providers and the users of the portal. In this section, we describe what representation formats underly the ontologies we use in our knowledge portals and the tools we use for constructing them.

Our domain ontologies consist of (i) concepts defining and structuring domain-specific terms, (ii) properties between concepts (i.e. relations) and between concepts and built-in types (i.e. attributes), and (iii) axioms that allow for additional inferences, such as the verification of constraints and the generation of new knowledge.

We model ontologies at an epistemological level using sophisticated graphical means of the ontology engineering workbench **OntoEdit** (cf. Figure 3 or [33] for a detailed description). The workbench offers different views for modeling concepts, attributes, relations and axioms. The resulting ontology may be translated into different actual representation languages, *viz.* F-Logic, RDF, OIL, and DAML-ONT (cf. `http://ontobroker.semanticweb.org/ontos/` for a number of ontologies in F-Logic, OIL and DAML-ONT; also cf. Table 1 on different ontology languages for the Web).

To illustrate the structure of the ontologies built with OntoEdit, the screenshot in Figure 3 depicts part of the KA2 ontology describing a research community as it is seen in the ontology development environment OntoEdit. The leftmost window depicts the is-a-relationship that structures the concepts of the domain in a taxonomy. Attributes and relations of concepts are inherited by subconcepts. Multiple inheritance is allowed as a concept may fit into different branches of the taxonomy. In Figure 3, attributes and relations of the concept AcademicStaff appear in the middle window. Some of these attributes, like FIRSTNAME and LASTNAME are inherited from the superordinate concept Person. Relations refer to other concepts, like WORKSATPROJECT denoting a relation between AcademicStaff and Project.

Beyond simple structuring we model axioms or rules, that are defined on top of the core ontology allowing inferencing and by that means the generation of new knowledge. For this purpose, we define semantic patterns [32] that describe generic reasoning behaviour. One example is the *common membership pattern*:

```
MembershipRelated(memberrelation,directrelation)
```

This pattern expresses that if two different instances i_1, i_2 belong to a set S via the membership relation `memberrelation`, they are related to each other via the `directrelation`.

[5] Cf. [34] about our methodology for constructing ontology-based applications.

[6] **Note for reviewers:**
currently the start address is: `http://aifbhermes.aifb.uni-karlsruhe.de/adl_demo/adl-frame.htm`, an indirection from `http://char.aifb.uni-karlsruhe.de/` will be provided soon. Transfer to a server without development activities scheduled for January 2001 will make it more stable for demonstration purposes before final printing of the paper.

Table 1: Overview of Ontology Languages and Systems on the Web

Quite a large number of representation languages for representing ontologies on the web have been established over the last decade. We here give a very brief survey of existing ontology representation languages and associated systems on the Web:[a]

The current starting point for ontology languages on the Web are recommendations of the W3C for representing semi-structured data on the Web with **Resource Description Framework (RDF)** [22] and for modeling concepts and relations with **RDF Schema (RDFS)** [39]. RDF represents the core data model that enables the encoding, exchange and reuse of semi-structured data, comprising a simple triple model for relations together with a convention for expressing reified facts and comes also with an XML style syntax. RDFS is an RDF application that basically allows to describe concept and property hierarchies as well as domain restrictions and range restrictions of properties. RDF and RDFS serves as a lightweight semantic layer that may be mapped onto other languages or that are used as a foundation for other languages.

OntoBroker [6] and our approach for building knowledge portals uses **F-Logic**, an object-oriented and logics-based representation language conceived by Kifer *et al.* [19]. It supports inferencing for query answering on schema and instance level extending horn logic with object-oriented primitives. In the implementation *SiLRI* (Simple Logic-based RDF Interpreter) by Angele and Decker [5] that we use, the F-Logic engine may integrate RDF and RDFS facts and reason on them.[b] In a similar category is **SHOE** (Simple HTML Ontology Extensions)[c] [16], which uses the PARKA knowledge representation system, that allows the user to define a frame-based ontology with class, subclass, and property links. Additionally, on top of this frame-based ontology, horn logic rules may be defined.

Conceptual Graphs (CGs) [30] are a system of logics based on the existential graphs of Charles Sanders Peirce and semantic networks. The WEB-KB system [25] describes an application similar to OntoBroker that embedds knowledge into web documents using conceptual graphs. A mapping of RDF into CGs is described in [4], where a CG mechanism is used to answer queries about stored RDF facts.

Description Logics are a fragment of first-order logic with rather expressive primitives but still decidable and (empirically) efficient inference procedures. LOOM [23] is a frequently used system with incomplete DL reasoning which has also been used commercially for Web applications. **OIL** (the ontology inference layer)[d] offers an integration of RDF/RDFS with a basically DL-based semantics [7]. It, thus, provides a semantically richer and more precise basis than RDF, embracing the current Web standards. The mapping of OIL, emerging into the efficient terminological reasoning system FaCT [17] is currently underway.

One of the most recent developments in the Darpa DAML intiative, the Darpa Agent Markup Language **DAML-ONT** has been proposed. As a layer on top of RDF/RDFS DAML-ONT[e] is — like OIL — intended to integrate ontologies with Web standards. Current efforts in DAML-ONT aim at integrating a rule language.

[a] Cf. [37] for an elaborated survey, which naturally cannot anymore reflect the current state of affairs.

[b] `http://www.ontoprise.de/download`

[c] `http://www.cs.umd.edu/projects/plus/SHOE/`

[d] http://www.ontoknowledge.org/oil

[e] http://www.daml.org/2000/10/daml-ont.html

Such a generic semantic pattern is instantiated by ontology concepts and/or relations through the graphical interface. For instance, "two persons that belong to a common project are said to collaborate" or "two persons that have written a common paper are co-authors".

```
MembershipRelated(worksAtProject,cooperatesWith).
MembershipRelated(writesPaper,coauthorOf).
```

This representation may then be (partially) translated into different target languages, as may be seen in Table 2.

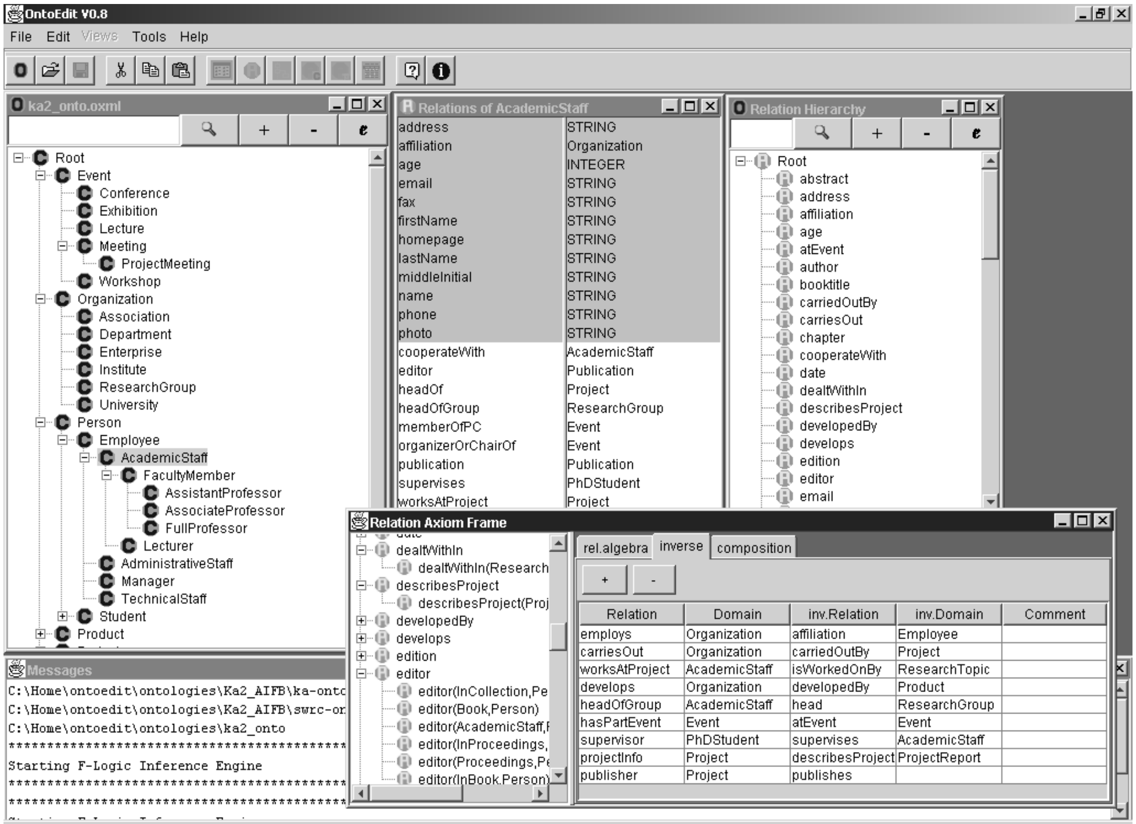

Figure 3: Part of the KA2 Ontology in an OntoEdit-Screenshot

6 Providing Knowledge

"One method fits all" does not meet the requirements we have sketched above for the information provisioning part of knowledge portals. What one rather needs is a set of methods and tools that may account for the diversity of information sources of potential interest for presentation at the knowledge portal. While these methods and tools need to obey different syntactic mechanisms, coherent integration of information is only possible with a conceptual basis that may sort loose pieces of information into a well-defined knowledge warehouse. In our setting, the conceptual basis is given through the ontology that provides the background knowledge and that supports the presentation of information by semantic, *i.e.* rule-enhanced queries. Talking about the syntactic and/or interface side, we support three major, different, modes of information provisioning: First, we handle *metadata-based information sources* that explicitly describe contents of documents on a semantic basis. Second, we align regularities found in documents or data structures with the corresponding semantic background knowledge in *wrapper-based* approaches. Thus, we may create a common conceptual denominator for previously unrelated pieces of information. Finally, we allow the direct provisioning and maintenance of facts through our *fact editor*. In addition to the mechanisms described above, we provide the developers of a knowledge portal with an RDF-based crawler, that searches the web with ontology focus for relevant instances described as RDF expressions. All the information is brought together in a knowledge warehouse that stores data and metadata alike. Thus, it mediates between the original information sources and the navigating and querying needs discussed in the next section.

Table 2: Resulting output for different ontology languages

Language	Result	Comment
F-Logic	FORALL x,y,z x[cooperatesWith->>y] <- x[worksAtProject->>z] and y[worksatProject->>z] and not equal(x,y).	
KIF	(=> (worksAtProject ?x ?z) (worksAtProject ?y ?z) (~= ?x ?y) (cooperatesWith ?x ?y))	
SHOE	<DEF-INFERENCE> <INF-IF> <RELATION NAME="worksAtProject"> <ARG POS=1 VAR VALUE="X"/> <ARG POS=2 VAR VALUE="Z"/> </RELATION> <RELATION NAME="worksAtProject"> <ARG POS=1 VAR VALUE="Y"/> <ARG POS=2 VAR VALUE="Z"/> </RELATION> </INF-IF> <INF-THEN> <RELATION NAME="cooperatesWith"> <ARG POS=1 VAR VALUE="X"/> <ARG POS=2 VAR VALUE="Y"/> </RELATION> </INF-THEN> </DEF-INFERENCE>	negation not allowed in SHOE, hence “partial” semantics incur overgeneration of “cooperatesWith” relationships

6.1 Metadata-based Information

Metadata-based information enriches documents with semantic information by explicitly adding metadata to the information sources. Over the last years several metadata languages have been proposed which can be used to annotate information sources. In our approach the specified ontology constitutes the conceptual backbone for the different syntactic mechanisms.

Current web standards for representing metadata like RDF [22] or XML [38] can be handled within our knowledge portal approach. We have developed a method and a tool called *DTDMaker* for generating document type definitions (DTDs) out of ontologies [10]. DTDMaker derives an XML document type definition from a given ontology, so that XML instances can be linked to an ontology. The linkage has the advantage that the document structure is grounded on a true semantic basis and, thus, facts from XML documents may be directly integrated into the knowledge warehouse. The method has the advantage that the large number of available XML tools, e.g. for editing documents become tools that provide formal metadata for the knowledge portals. HTML-A, early proposed by Fensel et al. [12], is an HTML extension which adds annotations to HTML documents using an ontology as a metadata schema. HTML-A has the advantage to smoothly integrate semantic annotations into HTML and prevents the duplication of information.

More wide-spread[7], RDF facts serve as direct input for the knowledge warehouse and RDF facts can be generated from information contained in the knowledge warehouse. An example of RDF metadata-based information is given through the following RDF expression, that states that the string "Rudi Studer" is the NAME of the instance of the concept FullProfessor with the object identifier `http://www.aifb.uni-karlsruhe.de/person:rst`. Additionally, the homepage of the object `http://www.aifb.uni-karlsruhe.de/person:rst` is defined via the attribute HOMEPAGE. These RDF facts are instantiated using the vocabulary given through the KA2 ontology.

```
<rdf:RDF
 xmlns:rdf  = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:ka2 = "http://www.semanticweb.org/ontologies/ka2-onto-2000-11-
07.rdfs"

  <ka2:FullProfessor rdf:ID="http://www.aifb.uni-karlsruhe.de/person:rst">
           <ka2:firstName>Rudi</ka2:name>
           <ka2:lastName>Studer</ka2:name>
           <ka2:homepage>
               http://www.aifb.uni-karlsruhe.de/Staff/studer.html
           </ka2:homepage>
 </ka2:FullProfessor>
</rdf:RDF>
```

To facilitate the annotation of HTML, we have developed an RDF-based annotation tool called *OntoAnnotate* (cf. Figure 4 where a merger between two IT companies, Gauss and Magellan, is captured). OntoAnnotate and its underyling mechanisms for semantic annotation are described in further detail in [9]. It is also possible to enrich documents generated with Microsoft Office applications with metadata by using our plug-ins *Word-RDF* and *Excel-RDF*.

For the future we envision a semi-automatic tool, which combines automatic information extraction techniques with manual accuracy. We currently do research on this task in the DAML OntoAgents project (`http://WWW-DB.Stanford.EDU/OntoAgents/`).

6.2 Wrapper-based Information

In general, annotating information sources by hand is a time consuming task. Often, however, annotation may be automated when one finds regularities in a larger number of documents. The principle idea behind wrapper-based information is that there are large information collections that have a similar structure. We here distinguish between semi-structured information sources (*e.g.* HTML) and structured information sources (*e.g.* relational databases).

[7] E.g., used for the `http://www.dmoz.org` directory.

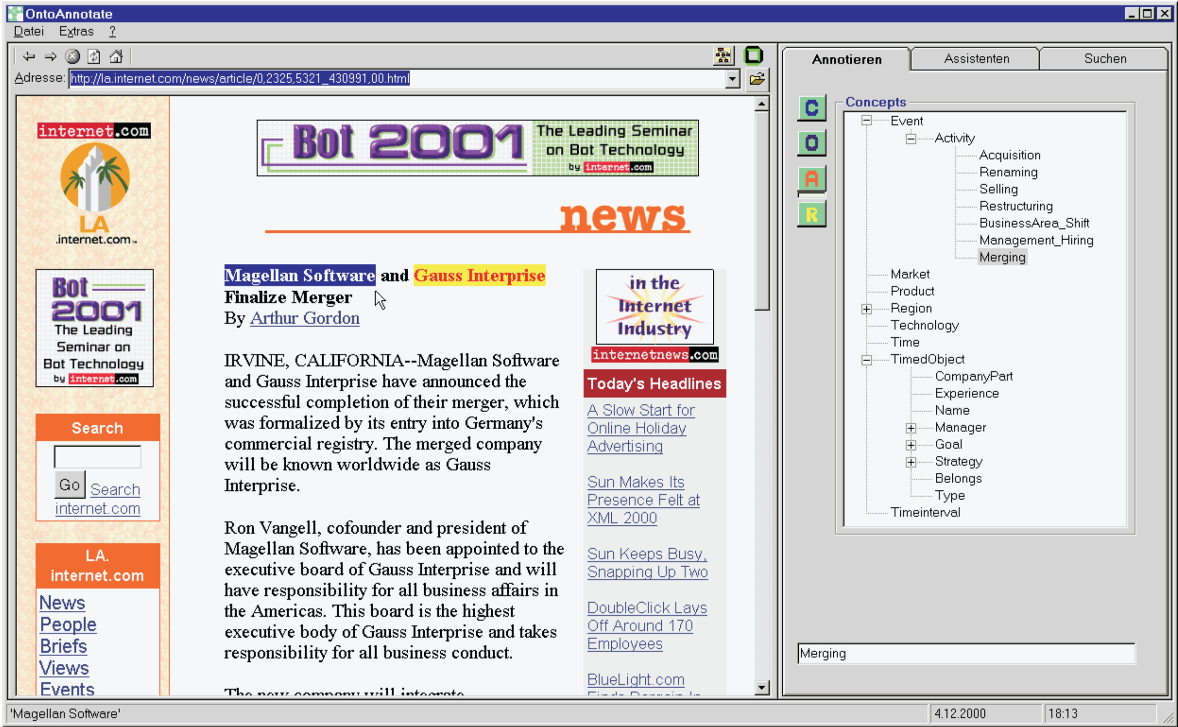

Figure 4: OntoAnnotate - Providing Semantics in HTML Documents

Semi-structured Sources. In recent years several approaches have been proposed for wrapping semi-structured documents, such as HTML documents. Wrapper factories (*cf.* Sahuguet et al. [29]) and wrapper induction (*cf.* Kushmerick [20]) have considerably facilitated the task of wrapper construction. In order to wrap directly into our knowledge warehouse we have developed our own wrapper approach that directly alignes regularities in semi-structured documents with their corresponding ontological meaning.

Structured Sources. Often existing databases and other legacy systems may contain valuable information for building a knowledge portal. Ontologies have shown their usefulness in the area of intelligent database integration. They act as information mediators (*cf.* Wiederhold & Genesereth [40]) between distributed and heterogeneous information sources and the applications that use these information sources. Existing entities in legacy systems are mapped onto concepts and relations defined in the ontology. Thus, existing information may be pumped into the knowledge warehouse by a batch process or it may be accessed on the fly.

6.3 Fact Editor.

The process of providing new facts into the knowledge warehouse should be as easy as possible. For this reason we offer the hyperbolic interface tool (*cf.* Figure 5) which may be used as a *Fact Editor*. In this mode its forms are not used to ask for values, but to insert values for attributes of instances of corresponding concepts from the ontology. The Fact Editor is also used for maintaining the portal, *viz.* to add, modify, or delete facts.

7 Access the Knowledge Portal

Having provided information with a conceptual underpinning, we now want to provide the same rich semantic structures to define a multitude of views that dynamically arrange information. Thus, our system may yield the kind of rich interlinking that is most adequate for the individual user and her navigation and querying of the knowledge portal. We start with a description of the query capabilities in our representation framework. While in principle we could use a number of different query languages, in practice our framework builds on the very same F-Logic mechanism for querying as it did for ontology representation and, thus, it may also exploit the ontological background knowledge. Through this semantic level we achieve the independence from the original, syntactically proprietary, information sources that we stipulated earlier. Nevertheless, F-Logic is as poorly suited for presentation to naive users as any other query language. Hence, its use is mostly disguised in various easy-to-use mechanisms that more properly serve the needs of the common user, while it still gives the editor all the power of the principal F-Logic representation and query capabilities.

7.1 Query Capabilities

To illustrate the range of queries employed in our portals, we here give a few simple examples. For instance, using a concrete example from our KA2 Portal the following query asks for all publications of the researcher with the last name “Studer”.

```
FORALL Pub <- EXISTS ResID
ResId:Researcher[lastName ->> "Studer";publication ->> Pub].
```

The substitutions for the variable *Pub* constitute the publications queried by this expression.

Besides retrieving explicit information, the query capabilities allow to make implicit information explicit. They use the background knowledge expressed in the domain ontology including rules as introduced above. If we have a look at web pages about research projects, information about the researchers (*e.g.* their names, their affiliation, ...) involved in the projects is often explicitly stated. However, the fact that researchers who are working together in projects are cooperating is typically left aside. A corresponding question might be: “Which researchers are cooperating with other researchers?” Querying for cooperating researchers the implicit information about project cooperation of researchers is exploited. The query may be formulated by:

```
FORALL ResID1,ResID2 <-
ResId1:Researcher[cooperatesWith ->> ResID2]
and ResID2:Researcher.
```

The result set includes explicit information about a researchers cooperation relationships, which are stored in the *knowledge warehouse*, and also implicit information about project cooperation between researchers derived using the project-cooperation rule modeled in the ontology and inferred by SiLRi.

Usually, it is too inconvenient for users to query the portal using F-Logic. Therefore we offer a range of techniques that allow for navigating and querying the knowledge portals we built:

- A *hypertext link* may contain a query that is dynamically evaluated when one clicks on the link. Browsing is made possible through the definition of views onto top-level concepts of the ontology, such as Persons, Projects, Organizations, Publications, Technology and

Organization. Each of these topics can be browsed using predefined views. For example, a click on the projects hyperlink results in a in a query for all projects known at the portal. The query is evaluated and the results are presented to the user in a table.

- A choice of concepts, instances, or combinations of both may be issued to the user in *HTML forms*. Choice options may be selected through check boxes, selection lists, or radio buttons. For instance, entering CHAR, an F-Logic query is evaluated and all existing companies contained in the portal are retrieved and dynamically offered for selecting among activities of companies in a drop down list. Search or selection may be further restricted using specific attributes contained in the ontology, such as more specific types of activities or shorter time periods.
- For the KA2Portal we have materialized the ontology with all its underlying facts (cf. `KBNavigate` in Figure 2). The ontology is offered in a tree view and a click on a concept directly shows all underlying instances.
- A query may also be generated by using the hyperbolic view interface (*cf.* Figure 5). The hyperbolic view visualizes the ontology as a hierarchy of concepts. The presentation is based on hyperbolic geometry (*cf.* [21]) where nodes in the center are depicted with a large circle, whereas nodes at the border of the surrounding circle are only marked with a small circle. This visualization technique allows a survey over all concepts, a quick navigation to nodes far away from the center, as well as a closer examination of nodes and their vicinity. When a user selects a node from the hyperbolic view, a form is presented which allows the user to select attributes or to insert values for the attributes. An example is shown in Figure 5. The user is searching for the community member "Studer" and his photo. Based on the selected node and the corresponding attributes, a query is compiled. The query-result is shown in the right part of Figure 2.
- Furthermore, queries created by the hyperbolic view interface may be stored using the personalization feature. Queries are personalized for the different users and are available for the user in a selection list. The stored queries can be considered as *semantic bookmarks*. By selecting a previously created bookmark, the underlying query is evaluated and the updated results are presented to the user. By this way, every user may create a personalized view onto the portal (cf. `Personalization` in Figure 2).
- Finally, we offer an expert mode. The most technical (but also most powerful and flexible) way for querying the portal requires that F-Logic is typed in by the user. This way is only appropriate for users who are very familiar with F-Logic and the domain ontology.

8 Conclusion

Knowledge portals serve as intermediaries for knowledge access and knowledge sharing on the Web. We have demonstrated how ontologies may lay a conceptual foundation, which supports the building of knowledge portals including means for knowledge access and provisioning. The two case studies that we have shown appear only as the tip of the iceberg of applications yet to come. Already now the first E-Commerce portals have started embracing ontologies, and corporate portals for managing enterprise internal knowledge are catching up [34]. Nevertheless, full-fledged support of ontology-based technology on the Web has been missing until now and also our approach needs to be extended in many directions, such as additional means for ontology-based personalization or log-mining with conceptual structures to name but two desiderata.

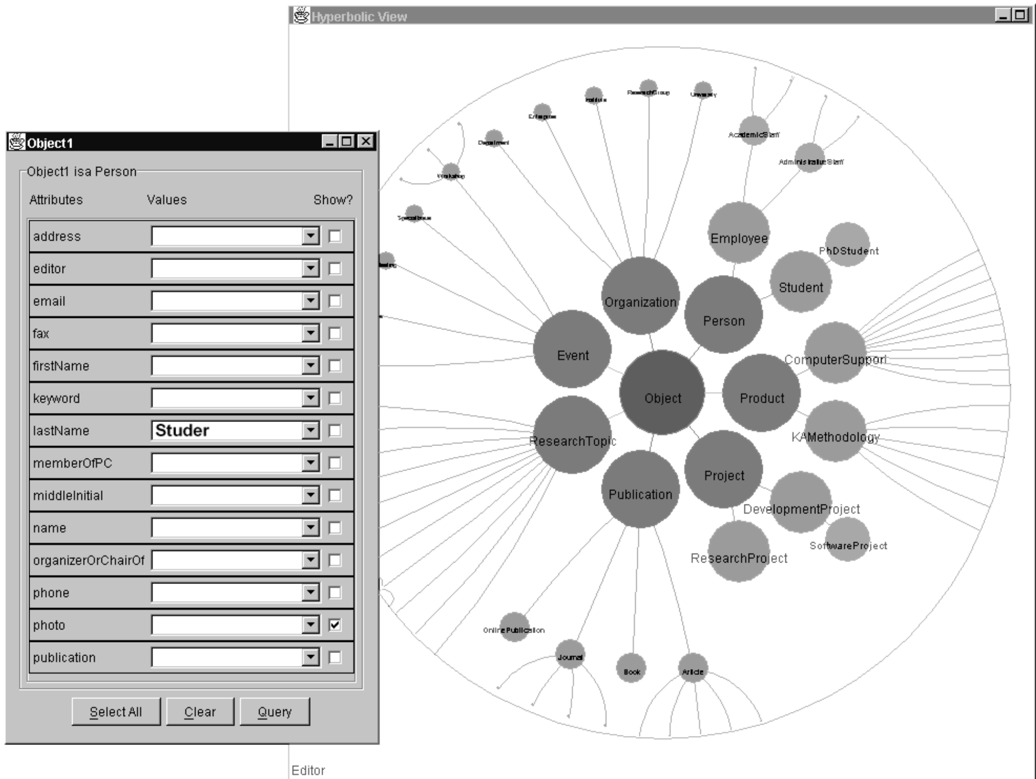

Figure 5: Hyperbolic Query View Interface

We think that our work on knowledge portals is only one very early starting point towards the Semantic Web, which will provide machine-readable information for all kinds of web-based applications. In particular, future applications will need to integrate more automatic techniques — for building ontologies [24], for providing meta data, and for learning from the usage of the Semantic Web.

Acknowledgements. The research presented in this paper would not have been possible without our colleagues or students at the Institute AIFB, University of Karlsruhe, and Ontoprise GmbH. We thank Jürgen Angele, Kalvis Apsitis (now: RITI Riga Information Technology Institute), Stefan Decker (now: Stanford University), Michael Erdmann, Dieter Fensel (now: VU Amsterdam), Siegfried Handschuh, Andreas Hotho, Mika Maier-Collin, Daniel Oberle, Hans-Peter Schnurr, Rudi Studer and York Sure. Research for this paper was partially financed by Ontoprise GmbH, Karlsruhe, Germany, by US Air Force in the DARPA DAML project "OntoAgents", by EU in the IST-1999-10132 project "On-To-Knowledge" and by BMBF in the project "GETESS" (01IN901C0).

References

[1] R.B. Altmann, M. Bada, X.J. Chai, M. Whirl Carillo, R.O. Chen, and N.F. Abernethy. RiboWeb: An Ontology-based System for Collaborative Molecular Biology. *IEEE Intelligent Systems*, 14(5):68–76, September/October 1999.

[2] J. Angele, H.-P. Schnurr, S. Staab, and R. Studer. The times they are a-changin' — the corporate history analyzer. In D. Mahling and U. Reimer, editors, *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management. Basel, Switzerland, October 30-31, 2000*, 2000. http://www.research.swisslife.ch/pakm2000/.

[3] R. Benjamins, D. Fensel, and S. Decker. KA2: Building Ontologies for the Internet: A Midterm Report. *International Journal of Human Computer Studies*, 51(3):687, 1999.

[4] O. Corby, R. Dieng, and C. Hebert. A conceptual graph model for w3c resource description framework. In *ICCS 2000 — International Conference on Conceptual Structures. Darmstadt, Germany, August 2000*, LNAI. Springer, 2000.

[5] S. Decker, D. Brickley, J. Saarela, and J. Angele. A Query and Inference Service for RDF. In *Proceedings of the W3C Query Language Workshop (QL-98), http://www.w3.org/TandS/QL/QL98/, Boston, MA, December 3-4*, 1998.

[6] S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In R. Meersman et al., editors, *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer Academic Publisher, 1999.

[7] S. Decker, D. Fensel, F. van Harmelen, I. Horrocks, S. Melnik, M. Klein, and J. Broekstra. Knowledge representation on the web. In *Proceedings of the 2000 International Workshop on Description Logics (DL2000), Aachen, Germany*, 2000.

[8] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. A Query Language for XML. In *Proceedings of the 8th Int. World Wide Web Conf. (WWW'8), Toronto, May 1999*, pages 1155–1169. Elsevier Science B.V., 1999.

[9] M. Erdmann, A. Maedche, H.-P. Schnurr, and Steffen Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *P. Buitelaar & K. Hasida (eds). Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, Luxembourg, August 2000.

[10] M. Erdmann and R. Studer. Ontologies as Conceptual Models for XML Documents. In *Proceedings of the 12th International Workshop on Knowledge Acquisition, Modelling and Mangement (KAW'99), Banff, Canada, October*, 1999.

[11] R. Faulstich. Internet portals for electronic commerce. Master's thesis, Institute AIFB, University of Karlsruhe, 2000. Download at http://www.aifb.uni-karlsruhe.de/ sst/Teaching/MastersTheses/closed.html.

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

[13] M. Fernandez, D. Florescu, J. Kang, and A. Levy. Catching the Boat with Strudel: Experiences with a Web-Site Management System. In *Proceedings of the 1998 ACM Int. Conf. on Management of Data (SIGMOD'98) , Seattle, WA*, pages 414–425, 1998.

[14] P. Fröhlich, W. Neijdl, and M. Wolpers. KBS-Hyperbook - An Open Hyperbook System for Education. In *10th World Conf. on Educational Media and Hypermedia (EDMEDIA'98), Freiburg, Germany*, 1998.

[15] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 6(2):199–221, 1993.

[16] J. Heflin and J. Hendler. Dynamic ontologies on the web. In *AAAI-2000 — Proceedings of the National Conference on Artificial Intelligence. Austin, TX, USA, August 2000*, 2000.

[17] I. Horrocks. Using an expressive description logic: Fact or fiction? In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998*, pages 636–649. Morgan Kaufmann, 1998.

[18] M. Kesseler. A Schema Based Approach to HTML Authoring. In *Proceedings of the 4th Int. World Wide Web Conf. (WWW'4). Boston, December*, 1995.

[19] M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42:741–843, 1995.

[20] N. Kushmerick. Wrapper Induction: Efficiency and Expressiveness. *Artificial Intelligence*, 118(1), 2000.

[21] L. Lamping, R. Rao, and P. Pirolli. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 401–408, 1995.

[22] O. Lassila and R. Swick. Resource description framework (RDF). model and syntax specification. Technical report, W3C, 1999. W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax.

[23] R. MacGregor. Inside the LOOM description classifier. *SIGART Bulletin*, 2(3):88–92, 1991.

[24] A. Maedche and S. Staab. Mining Ontologies from Text. In *Proceedings of EKAW-2000, Springer Lecture Notes in Artificial Intelligence (LNAI-1937), Juan-Les-Pins, France, 2000*. Springer, 2000.

[25] P. Martin and P. Eklund. Embedding Knowledge in Web Documents. In *Proceedings of the 8th Int. World Wide Web Conf. (WWW'8), Toronto, May 1999*, pages 1403–1419. Elsevier Science B.V., 1999.

[26] H. Maurer. *Hyperwave. The Next Generation Web Solution*. Addison Wesley, 1996.

[27] Ontology.Org and Chemdex.Org. Chemdex^{TM}, a vortex business. http://www.ontology.org/main/papers/cases/chemdex.html observed at Nov 24, 2000, 2000.

[28] J. Robie, J. Lapp, and D. Schach. XML Query Language (XQL). In *Proceedings of the W3C Query Language Workshop (QL-98) , Boston, MA, December 3-4*, 1998.

[29] A. Sahuguet and F. Azavant. Building Intelligent Web Applications Using Lightweight Wrappers. *to appear in: Data and Knowledge Engineering*, 2000.

[30] J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1992.

[31] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. Semantic community web portals. In *WWW9 — Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000*. Elsevier, 2000.

[32] S. Staab, M. Erdmann, and A. Maedche. Semantic patterns. Technical Report 412, Institute AIFB, University of Karlsruhe, 2000.

[33] S. Staab and A. Maedche. Ontology engineering beyond the modeling of concepts and relations. In V.R. Benjamins, A. Gomez-Perez, and N. Guarino, editors, *Proceedings of the ECAI-2000 Workshop on Ontologies and Problem-Solving Methods. Berlin, August 21-22, 2000*, 2000.

[34] S. Staab, H.-P. Schnurr, R. Studer, and Y. Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 2001. Special issue on Knowledge Management (to appear).

[35] B. Templin. Dethroning the content king. http://webreview.com/pub/1999/06/25/feature/content.html, 1999.

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

[37] Frank van Harmelen and Dieter Fensel. Practical knowledge representation for the web. In *Proceedings of the Workshop on Intelligent Information Integration (III99) during IJCAI-99, Stockholm, Sweden, August 1999*, 2000.

[38] W3C. XML Specification. http://www.w3.org/XML/, 1997.

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

[40] G. Wiederhold and M. Genesereth. The Conceptual Basis for Mediation Services. *IEEE Expert / Intelligent Systems*, 12(5):38–47, September/October 1997.