# Knowledge Portals
## Ontologies at Work

*Steffen Staab and Alexander Maedche*

■ Knowledge portals provide views onto domain-specific information on the World Wide Web, thus helping their users find relevant, domain-specific information. The construction of intelligent access and the contribution of information to knowledge portals, however, remained an ad hoc task, requiring extensive manual editing and maintenance by the knowledge portal providers. 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 study and one commercial case study that show how our approach, called SEAL (semantic portal), is used in practice.

Information on the World Wide Web is ubiquitous, but it is painful to find anything specific. Hence, services flourish that put up knowledge portals for a well-structured orientation on the web.[1] Although there are some general-purpose knowledge portals such as Yahoo, the majority of knowledge portals, however, are domain- or market-specific and serve a particular clientele, for example, Look-Look, which offers structured access to trends in youth culture for companies with an interest in this market.

Knowledge portals typically are maintained manually. Knowledge portal providers enlisten hoards of people to contribute information pieces that are shaped by human editors into many different views. The editors' problems include considerations about what information pieces are there, how to structure them, who should look at them, and who should provide them. However, the manual structuring and contribution of large amounts of information for easy access by the users becomes a difficult and expensive problem over time. Therefore, we have developed a method for building and maintaining knowledge portals, the key technology being ontologies to help structure, access, and provide information that has been aggregated by a collaboration of people. For this purpose, ontologies constitute the formal means that specify the domain of interest for the clientele of the knowledge portal (compare Gruber [1993]).

To date, ontologies have been used for research (Staab et al. 2000; Altman et al. 1999) and commercial purposes such as presenting and mediating information (Wiederhold and Genesereth 1997) on the web,[2] tackling intriguing parts of the overall problem of building and maintaining knowledge portals. This article presents a comprehensive concept for building and maintaining tasks, including the delivery of decentralized knowledge, as well as tools for accessing, warehousing, and inferring knowledge. We elaborate on the basic tools and methods; a case study serving research needs; and—briefly—a commercial portal currently under development that uses our approach.

## Requirements for Knowedge Portals

The aim of knowledge portals is to make knowledge accessible to users and allow users the exchange of knowledge. Knowledge portals specialize in a certain topic to offer deep coverage of the domain of interest and, thus, address a community of users. The portals are commonly built to include community services, such as online forums, mailing lists, and news articles of relevant guises (Faulstich 2000).

Even facing only a medium-size portal, the amount of information that is stored becomes extremely unwieldy to present and refind. In particular, the common categories, such as 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 becomes urgent. 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/she searches for knowledge to solve a problem or learn about a new topic.

In fact, a number of research proposals and commercial solutions exist that have recognized and approached this problem. For example, MATHNET4 introduces knowledge sharing for mathematicians through a database relying on Dublin Core metadata. Altman et al. (1999) 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 (for example, Martin and Eklund [1999]; Fernandez et al. [1998]; and Maurer [1996]), but a comprehensive concept for supporting the knowledge portal has been missing thus far. Such overall support has to include, of course, the graphic user interface (GUI) for accessing the portal contents and thereby addressing community-specific needs, but it also needs to consider the contribution of knowledge as well as the overall construction and maintenance of the portal.

## Knowledge Providing

An essential feature of a knowledge portal is the easy addition of new information and/or the easy updating of old information in a way such that it can easily be refound. Thus, information can 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 that information was provided originally. The knowledge portal must remain adaptable to the information sources contributed by its providers—not vice versa.

This requirement precludes the application of database-oriented approaches (for example, Maurer [1996]) because they presume that a uniform mode of storage exists that allows for the structuring of information 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 insufficient because their underlying transformation mechanisms (for example, XSLT or XQL [Deutsch et al. 1999; Robie, Lapp, and Schach 1998]) 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.

To integrate diverse information, we require another layer besides the common distinction into document, content, and layout, that is, explicit knowledge structures that can 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 (for example, HTML metatags), (2) manual provision of data to the knowledge repository, and (3) a range of different wrappers that encapsulate structured and semistructured information sources (for example, databases or HTML documents). The section entitled "Providing Knowledge" addresses these issues in detail.

## Knowledge Access

Navigating through a knowledge portal that is unknown is a rather difficult task in general. Information retrieval can facilitate the finding of pieces of text, but its use is not sufficient to provide novice users with the right means for exploring unknown terrain. This navigation 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 structured and comprehensive. Simple tree-structured portals can be easy to maintain, but the chance is extremely high that an inexperienced user looking for information gets stuck at a dead-end. 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 example, 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 home pages.

Here, we must face the trade-off between resources used for structuring the portal (money, humanpower) and the extent to which a comprehensive navigation structure can be provided. Because information in the knowledge portal will continually be expanded and updated, a richly interrelated presentation of information usually requires extensive editing, such as is done, for example, for Yahoo. In contrast, knowledge portals should exhibit comprehensive structuring of information virtually for free.

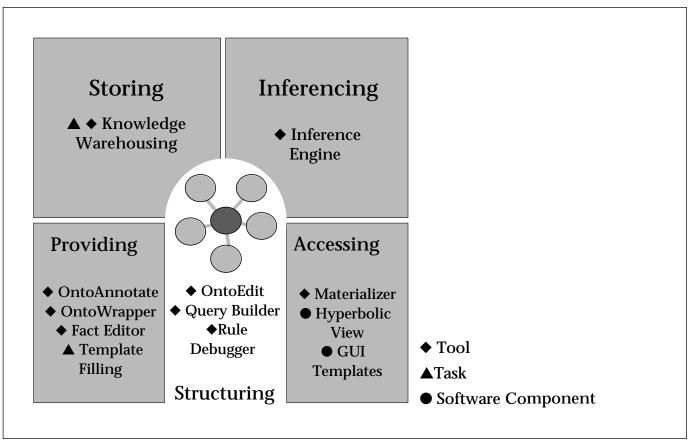Interesting research, for example, from Fröhlich, Neijdl, and Wolpers (1998) or Kesseler

*Figure 1. Knowledge Portal Architecture.*

(1995), demonstrates that authoring, as well as reading and understanding of web sites, profits from conceptual models underlying document structures *in the large*, that is, the interlinking between documents, as well as document structures *in the small,* that is, the contents of a particular document. In addition, it shows how rich linkage in multiple directions can be constructed automatically based on the underlying conceptual structures.

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 document structures in the large, comprehensive surveys and indexes of contents, and a large number of different views onto the contents of the portal, we require that the conceptual structure of the portal be made explicit at some point. We meet this requirement by providing an ontology. The section entitled Access the Knowledge Portal shows how conceptual structures are exploited for access purposes. However, first, we provide a high-level view onto the overall architecture (figure 1).

## Architecture

Our architecture is primarily divided 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 elaborate on these in subsequent sections; hence, we can safely ignore them here.

### Knowledge Warehousing

The knowledge warehouse hosts facts; metadata about documents; and the ontology, which describes the structure of the facts and the metadata. 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, therefore, is flexible with regard to changes and extensions 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 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 we describe, we could

restrict our attention to incoming data that were 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 resource description framework [RDF] interpreter) described in Decker et al. (1998). Basically, SILRI offers representation capabilities for RDF and F-LOGIC and combinations thereof. RDF is a frame-oriented representation language with an XML syntax. F-LOGIC is an object-oriented logic mechanism that extends datalog with object-oriented modeling primitives. Although RDF only allows for the contribution of facts and concept definitions, F-LOGIC also allows the querying and use of axioms.

For our purpose, SILRI is ideally suited because it allows for the combined querying of facts and ontological concepts. Hence, one can make statements such as "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 by the inference engine, thus offering a uniform mode of access. However, the inference engine caches previous queries to deliver short response times.

### Structuring

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

## Case Study: KA2 Portal

The first knowledge portal that we constructed was for the Knowledge Annotation Initiative of the knowledge-acquisition community (KA2) (compare Benjamins, Fensel, and Decker [1999]). The KA2 initiative was conceived for semantic knowledge retrieval from the web, building on knowledge created in the knowledge-acquisition community. To structure knowledge, an ontology has been built by an international collaboration of researchers. The ontology constitutes the basis for annotating web documents from the knowledge-acquisition community to enable intelligent access to these documents and 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 article. Some views of the KA2 contents can be seen in our up and running demonstration KA2 community web portal (figure 2).[3]

## 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 knowledgeable 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 subsegment X, or what are peer groups of companies in sector X.

For example, venture capitalists are given a large number of business proposals that they must decide on quickly about whether to invest. Typically, venture capitalists are experts in financial issues of starting a company, and accordingly, they use their financial expertise as a sieve to sort out the good potential investments. 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 be too expensive. From financial and technical points of view, evaluation criteria of different grain sizes are used that take up different amounts of time and money. A successful proposal would run through several evaluation cycles where increasingly fine-grained criteria and increasingly time-consuming evaluation measures are applied to sort out the good potential investments from the bad.

From the outsourcing of technical expertise comes a difficult problem: The duration between proposal and answer is rather long as the evaluation goes through several stages. Thus, investors clog their working line, and more importantly, they can miss good chances because proposers can 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.

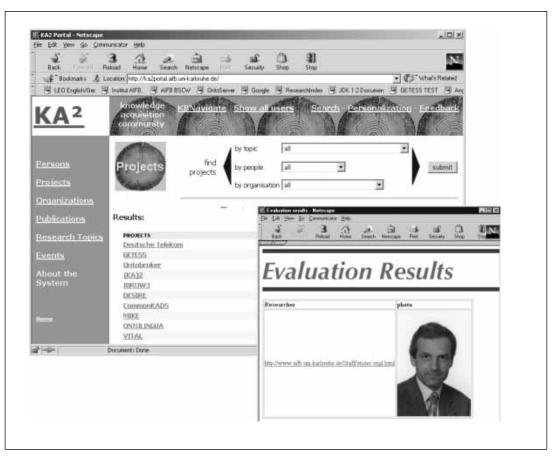The TIME2RESEARCH knowledge portal aims at streamlining the process that the technical

*Figure 2. KA2 Portal User Interface.*

analyst performs because it allows for collaborative knowledge contribution. The portal optimizes the information-delivery process between the venture capitalist and the technical expert because it allows for decentralized knowledge querying. It allows a 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 can answer his/her standard questions to the portal in a few minutes instead of triggering a day- or week-long process.[4]

## Structuring the Knowledge Portal

Ontologies have been established for knowledge sharing and are used as a means for conceptually structuring domains of interest (Wiederhold and Genesereth 1997; Uschold and King 1995). Because knowledge portals focus on particular domains, ontologies appear ideally suited to support knowledge sharing and reuse 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 (1) concepts defining and structuring domain-specific terms; (2) properties between concepts (that is, relations) and between concepts and built-in types (that is, attributes); and (3) 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 the sophisticated graphic means of the ontology engineering workbench ONTOEDIT (figure 3) (Staab and Maedche 2000).[5] The workbench offers different views for modeling concepts, attributes, relations, and axioms. The resulting ontology can be translated into different actual representation languages, that is, F-LOGIC, RDF, OIL, and DAML + OIL (table 1).[6]
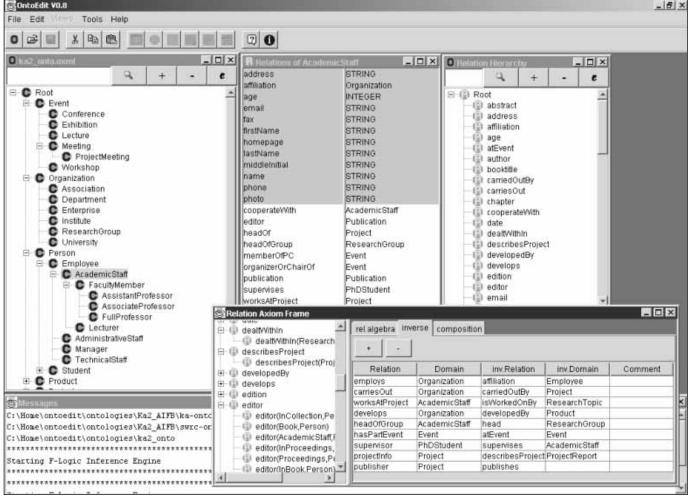
*Figure 3. Part of the KA2 Ontology in an ONTOEDIT Screenshot.*

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 because a concept might 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, such as FirstName and LastName, are inherited from the superordinate concept Person. Relations refer to other concepts, such as WorksAtProject denoting a relation between AcademicStaff and Project.

Beyond simple structuring, we model axioms or rules, which are defined on top of the core ontology allowing inferencing and, thus, the generation of new knowledge. For

this purpose, we define semantic patterns (Staab, Erdmann, and Maedche 2001) that describe generic reasoning behavior. 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 $S$ by way of the membership relation memberrelation, they are related to each other by the directrelation.

Such a generic semantic pattern is instantiated by ontology concepts or relations through the graphic interface. For example, "two persons that belong to a common project are said to collaborate" or "two persons that have written a common paper are coauthors."

MembershipRelated(worksAtProject, cooperatesWith).

MembershipRelated(writesPaper, coauthorOf).

This representation can then be (partially)

translated into different target languages (compare table 1 for a survey), as can be seen in table 2.

# Providing Knowledge

"One method fits all" does not meet the requirements we have sketched here for the information contribution part of knowledge portals. What one rather needs is a set of methods and tools that can account for the diversity of information sources of potential interest for presentation at the knowledge portal. Although these methods and tools need to obey different syntactic mechanisms, coherent integration of information is only possible with a conceptual basis that can 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, that is, rule-enhanced queries. Talking about the syntactic or interface side, we support three major, different modes of information contribution: 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 can create a common conceptual denominator for previously unrelated pieces of information. Third, we allow the direct contribution and maintenance of facts through our fact editor. In addition to the mechanisms described earlier, 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.

## 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 that 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 such as RDF (Lassila and Swick 1999)

Quite a large number of representation languages for representing ontologies on the web have been established over the last decade. Here, we here give a brief survey of existing ontology representation languages and associated systems on the web:[1]

The current starting point for ontology languages on the web are recommendations of the W3C for representing semistructured data on the web with resource description framework (RDF) and for modeling concepts and relations with RDF schema (RDFS).[2,3] RDF represents the core data model that enables the encoding, exchange, and reuse of semistructured data, comprising a simple triple model for relations together with a convention for expressing reified facts, and also comes with an XML-style syntax. RDFS is an RDF application that basically allows you to describe concept and property hierarchies as well as domain restrictions and range restrictions of properties. RDF and RDFS serve as a lightweight semantic layer that can be mapped onto other languages or that are used as a foundation for other languages.

ONTOBROKER (Decker et al. 1999) and SEAL (semantic portal), our approach for building knowledge portals, use F-LOGIC, an object-oriented and logics-based representation language conceived by Kifer, Lausen, and Wu (1995). It supports inferencing for query answering on schema and instance level, extending horn logic with object-oriented primitives. In the implementation SiLRI by Angele and Decker (Decker et al. 1998) that we use, the F-LOGIC engine can integrate RDF and RDFS facts and reason on them.[4] In a similar category is SHOE (simple HTML ontology extensions) (Heflin and Hendler 2000)[5], which uses the PAR-KA knowledge representation system, allowing 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 can be defined.

Conceptual graphs (Sowa 1992) are a system of logics based on the existential graphs of Charles Sanders Peirce and semantic networks. The WEB-KB system (Martin and Eklund 1999) describes an application similar to ONTOBROKER that embeds knowledge in web documents using conceptual graphs. A mapping of RDF into conceptual graphs is described in Corby, Dieng, and Hebert (2000), where a conceptual graph 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 (MacGregor 1991) is a frequently used system with incomplete description logic reasoning that has also been used commercially for web applications. OIL (the ontology inference layer) offers an integration of RDF-RDFS with basically a description logic–based semantics (Decker et al 2000).[6] Thus, it provides a semantically richer and more precise basis than RDF, embracing the current web standards. There is a mapping of OIL into the efficient terminological reasoning system FACT (Horrocks 1998).

One of the most recent developments in the Defense Advanced Research Projects Agency (DARPA) Agent Markup Language (DAML) Initiative is the proposal of the language DAML-ONT. As a layer on top of RDF/RDFS, DAML-ONT is—like OIL—intended to integrate ontologies with web standards.[7] Current efforts in DAML aim toward the integration with OIL into DAML + OIL as well as toward the integration of a rule language.

1. Compare van Harmelen and Fensel (1999) for an excellent survey, which naturally cannot anymore cover the current state of affairs completely.

2. W3C. RDF Schema Specification. www.w3.org/TR/PR-rdf-schema/.

3. W3C Recommendation available at www.w3.org/TR/REC-rdf-syntax.

4. www.ontoprise.de/download.

5. www.cs.umd.edu/projects/plus/SHOE/.

6. www.ontoknowledge.org/oil.

7. www.daml.org/2000/10/daml-ont.HTML.

*Table1. Overview of Ontology Languages and Systems on the Web.*

```
<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:firstName>
   <ka2:lastName>Studer</ka2:lastName>
   <ka2:homepage
rdf:resource="http://www.aifb.uni-karlsruhe.de/Staff/studer.html"/>
   </ka2:FullProfessor>
</rdf:rdf>
```

*Figure 4. RDF Facts Instantiated Using the*
*Vocabulary Given through the KA2 Ontology.*

| 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 | | negation not allowed in SHOE, hence "partial" semantics incur overgeneration of "cooperatesWith" relationships |
| | <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> | |

*Table 2. Resulting Output for Different Ontology Languages.*
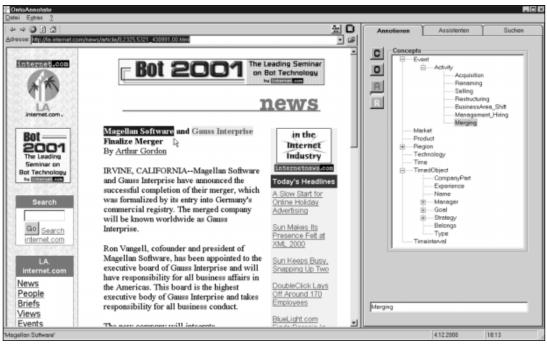
*Figure 5. ONTOANNOTATE—Providing Semantics in HTML Documents.*

or XML can be handled within our knowledge portal approach.[7] We have developed a method and a tool called DTDMAKER for generating document-type definitions (DTDs) out of ontologies (Erdmann and Studer 1999). DTD-MAKER derives an XML DTD from a given ontology so that XML instances can be linked to an ontology. The link has the advantage of grounding the document structure on a true semantic basis; thus, facts from XML documents can be integrated directly into the knowledge warehouse. The method has the advantage of having the large number of available XML tools, for example, for editing documents, become tools that provide formal metadata for the knowledge portals. HTML-A, early proposed by Fensel et al. (1998), is an HTML extension that adds annotations to HTML documents using an ontology as a metadata schema. HTML-A has the advantage of smoothly integrating semantic annotations into HTML and preventing the duplication of information.

More widespread, 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, which states that the string *Rudi Studer* is the Name of the instance of the concept FullProfessor with the object identifier www.aifb.uni-karlsruhe.de/person:rst. Additionally, the home page of the object www.aifb.uni-karlsruhe.de/person:

rst is defined by the attribute Homepage. These RDF facts are instantiated using the vocabulary given through the KA2 ontology (figure 4).

To facilitate the annotation of HTML, we have developed an RDF-based annotation tool called ONTOANNOTATE (compare figure 5, where a merger between two information technology companies, Gauss and Magellan, is captured). ONTOANNOTATE and its underlying mechanisms for semantic annotation are described in further detail in Erdman et al. (2000). 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 semiautomatic tool that combines automatic information-extraction techniques with manual accuracy. We currently do research on this task as part of the DAML ONTOAGENTS Project.[8]

## Wrapper-Based Information

In general, annotating information sources by hand is a time-consuming task. Often, however, annotation can 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 semistructured information sources (for example, HTML) and structured information sources (for example, relational databases).
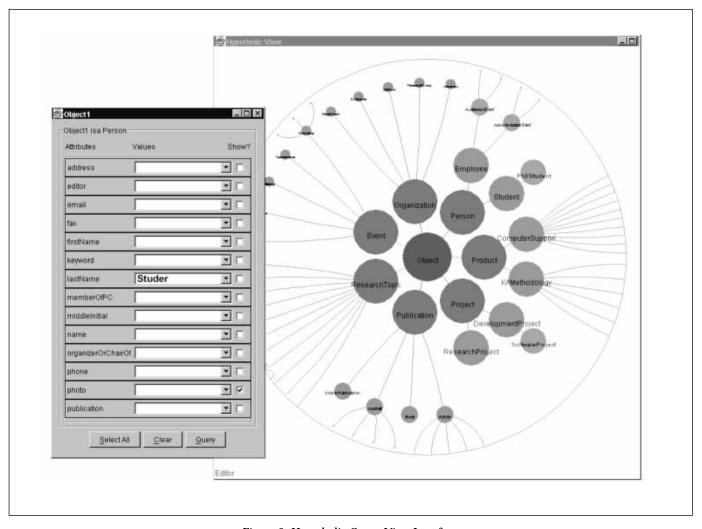
*Figure 6. Hyperbolic Query View Interface.*

**Semistructured Sources**

In recent years, several approaches have been proposed for wrapping semistructured documents, such as HTML documents. Wrapper factories (compare Sahuguet and Azavant [2001]) and wrapper induction (compare Kushmerick [2000]) have considerably facilitated the task of wrapper construction. To wrap directly into our knowledge warehouse, we have developed our own wrapper approach that directly aligns regularities in semistructured documents with their corresponding ontological meaning.

**Structured Sources**

Often, existing databases and other legacy systems can 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 (compare Wiederhold and

Genesereth [1997]) 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 can be pumped into the knowledge warehouse by a batch process, or it can be accessed on the fly.

**Fact Editor**

The process of providing new facts for the knowledge warehouse should be as easy as possible. For this reason, we offer the *hyperbolic interface tool* (compare figure 6) that can 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, that is, to add, modify, or delete facts.

## 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 can yield the kind of rich interlinking that is most adequate for the individual user and his/her navigation and querying of the knowledge portal. We start with a description of the query capabilities in our representation framework. Although 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; thus, it can 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, although it still gives the editor all the power of the principal F-LOGIC representation and query capabilities.

### Query Capabilities

To illustrate the range of queries used in our portals, we give a few simple examples. For example, 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 implicit information to be made explicit. They use the background knowledge expressed in the domain ontology, including rules as introduced earlier. If we have a look at web pages about research projects, information about the researchers (for example, their names and 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? When querying for cooperating researchers, the implicit information about project cooperation of researchers is exploited. The query can be formulated as

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

The result set includes explicit information about a researcher's 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 can 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 searched using predefined views. For example, a click on the projects hyperlink results 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 can be issued to the user in HTML forms. Choice options can be selected through check boxes, selection lists, or radio buttons. For example, 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 can be further restricted using specific attributes contained in the ontology, such as more specific types of activity or shorter time periods.

For the KA2 portal, we have materialized the ontology with all its underlying facts (compare 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 can also be generated by using the hyperbolic view interface (compare figure 5). The hyperbolic view visualizes the ontology as a hierarchy of concepts. The presentation is based on hyperbolic geometry (compare Lamping, Rao, and Pirolli [1995]), 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, and a closer examination of nodes and their vicinity. When a user selects a node from the hyperbolic view, a form is presented that allows the user to select attributes

or 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 can 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. Thus, every user can create a personalized view onto the portal (compare 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 be typed in by the user. This way is only appropriate for users who are very familiar with F-LOGIC and the domain ontology.

## Conclusion

Knowledge portals serve as intermediaries for knowledge access and knowledge sharing on the web. We have demonstrated how ontologies can lay a conceptual foundation that supports the building of knowledge portals, including means for knowledge access and contribution. The two case studies that we showed appear only as the tip of the iceberg of applications yet to come. Already now, the first electronic-commerce portals have started embracing ontologies, and corporate portals for managing enterprise internal knowledge are catching up (Staab et al. 2001). Nevertheless, full-fledged support of ontology-based technology on the web has been missing until now, and our approach needs to be extended in many directions, such as additional means for ontology-based personalization or log mining with conceptual structures.

We think that our work on knowledge portals is only one very early starting point toward the semantic web that 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 (Maedche and Staab 2000), providing metadata, and learning from the use of the semantic web.

## Acknowledgments

## Notes

1. B. Templin, 1999, Dethroning the Content King, available at webreview.com/pub/1999/06/25/feature/content.HTML.

2. Chemdex—A Vortex Business. www.ontology.org/main/papers/cases/chemdex.HTML.

3. ka2portal.aifb.uni-karlsruhe.de.

4. A demo version of the portal is set up during the printing of this article at www.time2research.de/.

5. A simplified version of ONTOEDIT is available for download at www.ontoprise.com.

6. Compare ontobroker.semanticweb.org/ontos/ for a number of ontologies in F-LOGIC, OIL, and DAML-ONT; also compare table 1 on different ontology languages for the web.

7. W3C. XML Specification. www.w3.org/XML/.

8. WWW-DB.Stanford.EDU/OntoAgents/.

## References

Altman, R. B.; Bada, M.; Chai, X. J.; Whirl Carillo, M.; Chen, R. O.; and Abernethy, N. F. 1999. RIBOWEB: An Ontology-Based System for Collaborative Molecular Biology. *IEEE Intelligent Systems* 14(5): 68–76.

Angele, J.; Schnurr, H.-P.; Staab, S.; and Studer, R. 2000. The Times They Are A-Changin'—The Corporate History Analyzer. Paper presented at the Third International Conference on Practical Aspects of Knowledge Management, 30–31 October, Basel, Switzerland.

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

Corby, O.; Dieng, R.; and Hebert, C. 2000. A Conceptual Graph Model for W3C Resource Description Framework. In *Proceedings of the ICCS 2000—International Conference on Conceptual Structures.* Lecture Notes in Artificial Intelligence, 468–482. Berlin: Springer-Verlag.

Decker, S.; Brickley, D.; Saarela, J.; and Angele, J. 1998. A Query and Inference Service for RDF. Paper presented at the W3C Query Language Workshop (QL-98), 3–4 December, Boston, Massachusetts.

Decker, S.; Erdmann, M.; Fensel, D.; and Studer, R. 1999. ONTOBROKER: Ontology-Based Access to Distributed and Semi-Structured Information. In *Database Semantics: Semantic Issues in Multimedia Systems,* eds. R. Meersman, Z. Tari, and S. M. Stevens, 351–369. New York: Kluwer Academic.

Decker, S.; Fensel, D.; van Harmelen, F.; Horrocks, I.; Melnik, S.; Klein, M.; and Broekstra, J. 2000. Knowledge Representation on the Web. Paper presented at the 2000 International Workshop on Description Logics (DL2000), 17–19 August, Aachen, Germany.

Deutsch, A.; Fernandez, M.; Florescu, D.; Levy, A.; and Suciu, D. 1999. A Query Language for XML. In *Proceedings of the Eighth International World Wide Web Conference (WWW'8),* 1155–1169. New York: Elsevier.

Erdmann, M., and Studer, R. 1999. Ontologies as Conceptual Models for XML Documents. Paper presented at the Twelfth International Workshop on Knowledge Acquisition, Modeling, and Management (KAW'99), 16–21 October, Banff, Canada.

Erdmann, M.; Maedche, A.; Schnurr, H.-P.; and Staab, S. 2000. From Manual to Semiautomatic Semantic Annotation: About Ontology-Based Text Annotation Tools. Paper presented at the COLING 2000 Workshop on Semantic Annotation and Intelligent Content, 5–6 August, Luxembourg.

Faulstich, R. 2000. Internet Portals for Electronic Commerce. Master's thesis, Institute for Applied Informatics and Formal Description Methods, University of Karlsruhe.

Fensel, D.; Decker, S.; Erdmann, M.; and Studer, R. 1998. ONTOBROKER: The Very High Idea. Paper presented at the Eleventh International Flairs Conference (FLAIRS-98), 1–5 May, Sanibel Island, Florida.

Fernandez, M.; Florescu, D.; Kang, J.; and Levy, A. 1998. Catching the Boat with STRUDEL: Experiences with a Web-Site Management System. In Proceedings of the 1998 ACM International Conference on Management of Data (SIGMOD'98), 414–425. New York: Association of Computing Machinery.

Fröhlich, P.; Neijdl, W.; and Wolpers, M. 1998. KBS-HYPERBOOK—An Open-Hyperbook System for Education. Paper presented at the Tenth World Conference on Educational Media and Hypermedia (EDMEDIA'98), 20–21 June, Freiburg, Germany.

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

Heflin, J., and Hendler, J. 2000. Dynamic Ontologies on the Web. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 443–449. Menlo Park, Calif.: American Association for Artificial Intelligence.

Horrocks, I. 1998. Using an Expressive Description Logic: Fact or Fiction? In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98),* 636–649. San Francisco, Calif.: Morgan Kaufmann.

Kesseler, M. 1995. A Schema-Based Approach to HTML Authoring. Paper presented at the Fourth International World Wide Web Conference (WWW'4), 11–14 December, Boston, Massachusetts.

Kifer, M.; Lausen, G.; and Wu, J. 1995. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM* 42(4): 741–843.

Kushmerick, N. 2000. Wrapper Induction: Efficiency and Expressiveness. *Artificial Intelligence* 118(1): 15–68.

Lamping, L.; Rao, R.; and Pirolli, P. 1995. 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, 401–408. New York: Association of Computing Machinery.

Lassila, O., and Swick, R. 1999. Resource Description Framework (RDF). Model and Syntax Specification. Technical Report, W3C, Boston, Massachusetts.

MacGregor, R. 1991. Inside the LOOM Description Classifier. *SIGART Bulletin* 2(3): 88–92.

Maedche, A., and Staab, S. 2001. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* (Special Issue on Semantic Web) 16(2).

Martin, P., and Eklund, P. 1999. Embedding Knowledge in Web Documents. In *Proceedings of the Eighth International World Wide Web Conference (WWW'8),* 1403–1419. New York: Elsevier Science.

Maurer, H. 1996. *Hyperwave: The Next Generation Web Solution.* Reading, Mass.: Addison Wesley.

Robie, J.; Lapp, J.; and Schach, D. 1998. XML Query Language (XQL). Paper presented at the W3C Query Language Workshop (QL-98), 3–4 December, Boston, Massachusetts.

Sahuguet, and Azavant, F. 2001. Building Intelligent Web Applications Using Lightweight Wrappers. *Data and Knowledge Engineering* (Special Issue on Intelligent Information Integration) 36(3): 283–316.

Sowa, J. F. 1992. *Conceptual Structures: Information Processing in Mind and Machine.* Reading, Mass.: Addison-Wesley

Staab, S., and Maedche, A. 2000. Ontology Engineering beyond the Modeling of Concepts and Relations. Paper presented at the ECAI-2000 Workshop on Ontologies and Problem-Solving Methods, 21–22 August, Berlin, Germany.

Staab, S.; Erdmann, M.; and Maedche, A. 2001. Engineering Ontologies Using Semantic Patterns. Paper presented at the IJCAI-2001 Workshop on E-Business and the Intelligent Web, 5 August, Seattle, Washington.

Staab, S.; Schnurr, H.-P.; Studer, R.; and Sure, Y. 2001. Knowledge Processes and Ontologies. *IEEE Intelligent Systems* (Special Issue on Knowledge Management) 16(1): 26–35.

Staab, S.; Angele, J.; Decker, S.; Erdmann, M.; Hotho, A.; Maedche, A.; Schnurr, H.-P.; Studer, R.; and Sure, Y. 2000. Semantic Community Web Portals. In *WWW9/Computer Networks* (Special Issue: WWW9—Proceedings of the Ninth International World Wide Web Conference, Amsterdam, The Netherlands, May, 15–19, 2000) 33(1–6): 473–491.

Uschold, M., and King, M. 1995. Toward a Methodology for Building Ontologies. Paper presented at the Workshop on Basic Ontological Issues in Knowledge Sharing, Fourteenth International Joint Conference on Artificial Intelligence, 20–25 August, Montreal, Canada.

van Harmelen, F., and Fensel, D. 1999. Practical Knowledge Representation for the Web. Paper presented at the Workshop on Intelligent Information Integration (III99), Sixteenth International Joint Conference on Artificial Intelligence, 31 July–6 August, Stockholm, Sweden.

Wiederhold, G., and Genesereth, M. 1997. The Conceptual Basis for Mediation Services. *IEEE Intelligent Systems* 12(5): 38–47.

**Steffen Staab** received an M.S.E. from the University of Pennsylvania in 1994 and a Dr. rer.nat. from the University of Freiburg in 1998, both in informatics. After consulting with the Fraunhofer Institute for Industrial Engineering, Stuttgart, he joined the University of Karlsruhe, where he is now an assistant professor. In 1999, he cofounded Ontoprise GmbH, a company providing a wide range of technologies centering on ontologies. Staab has been working and publishing in computational linguistics, text mining, knowledge management, ontologies, and the semantic web. He won Best Paper Award for a paper on constraint reasoning at ECAI-1998, and he is chairing the Semantic Web Workshop in Hong Kong at WWW 10. His e-mail address is sst@aifb.uni-karlsruhe.de.

**Alexander Maedche** is a Ph.D. student at the Institute of Applied Informatics and Formal Description Methods, University of Karlsruhe. In 1999, he received a diploma in industrial engineering, majoring in computer science and operations research, also from the University of Karlsruhe. His diploma thesis on knowledge discovery earned him a Best Thesis Award at the University of Karlsruhe. Maedche`s research interests cover knowledge discovery in data and text, ontology engineering, learning and application of ontologies, and the semantic web. Recently, he started to build a new research group at the FZI Research Center for Information Technologies at the University of Karlsruhe that researches semantic web technologies and applies them to knowledge management applications in practice. His e-mail address is ama@aifb.uni-karlsruhe.de.