ONTOBROKER: Ontology based Access to Distributed and Semi-Structured Information

Stefan Decker, Michael Erdmann, Dieter Fensel and Rudi Studer
University of Karlsruhe, Institute AIFB, D-76128 Karlsruhe, Germany
((decker, erdmann, fensel, studer)@aifb.uni-karlsruhe.de*)*
http://www.aifb.uni-karlsruhe.de/WBS/broker

Abstract. The World Wide Web (WWW) can be viewed as the largest multimedia database that has ever existed. However, its support for query answering and automated inference is very limited. Metadata and domain specific ontologies were proposed by several authors to solve this problem. We developed Ontobroker which uses formal ontologies to extract, reason, and generate metadata in the WWW. The paper describes the formalisms and tools for formulating queries, defining ontologies, extracting metadata, and generating metadata in the format of the Resource Description Framework (RDF), as recently proposed by the World Wide Web Consortium (W3C). These methods provide a means for semantic based query handling even if the information is spread over several sources. Furthermore, the generation of RDF descriptions enables the exploitation of the ontological information in RDF-based applications.

1. Introduction

In more and more application areas large collections of digitized multimedia information are gathered and have to be maintained (e.g. in medicine, chemical applications or product catalogs). Therefore, there is an increasing demand for tools and techniques supporting the management and usage of digital multimedia data. Especially the World Wide Web (WWW) can be regarded as the largest multimedia database that ever existed and every day more and more data is available through it. Its support for retrieval and usage is very limited because its main retrieval services are keyword-based search facilities carried out by different search engines, web crawlers, web indices, man-made web catalogs etc. Given a keyword, such services deliver a set of pages from the web that use this keyword. *Ontologies* and metadata (based on ontologies) are proposed as a means for retrieving and using multimedia data (Boll et al. 98) (Sheth&Klas 98). They provide "an explicit specification of a conceptualization" (Gruber 93) and are discussed in the literature as means to support knowledge sharing and reuse (Farquhar et al. 97) (Fridman Noy&Hafner 97). This approach to reuse is based on the assumption that if a modeling scheme – i.e. an ontology— is explicitly specified and agreed upon by a number of agents, it is then possible for them to share and reuse knowledge. Clearly, it is unlikely that there will be a common ontology for the whole population of the WWW and every subject. This leads to the *metaphor of a newsgroup* or *domain specific ontology* (Kashyap&Shet 97) (Mena et al. 98) to define the terminology for a group of people which share a common view on a specific domain. Using ontologies for information retrieval has certain advantages over simple keyword based access methods: An ontology provides a shared vocabulary for expressing information about the contents of (multimedia) documents. In addition, it includes axioms for specifying relationships between concepts. Such an ontology may then in turn be used to formulate semantic queries and to deliver exactly the information we are interested in. Furthermore, the axioms provide a means for deriving information which has been specified only implicitly.

These advantages come with the price of having to provide information in a more formal manner. Since a large portion of the WWW is formulated using HTML, which is not an entirely formal language, the following questions arise:

- How can information be represented (in a sufficiently formal way) in the WWW?
- How can this information be extracted and maintained in the WWW?
- How can we reason with it and what inferences are possible?

To answer the first question, we have to look at the effort toward standardizing data, metadata, and ontologies. XML based languages (XML) are becoming standard formats for representing data in the WWW (even for multimedia data, see e.g. Precision Graphics Markup Language (PGML) or the Synchronized Multimedia Integration Language (SMIL 1.0). Based on XML, the metadata standard RDF (Resource Description Framework (RDF)) and the RDF schema language (RDFS), which can be used to express ontologies, are under development and will probably be widely used in the near future. The use of these standards allows to access a variety of data in the WWW in a more formal way than today.

For answering the other two questions, we developed a system called Ontobroker (Fensel et al. 98a) (Ontobroker) with the following core elements (see Figure 1):

- The most central part are the ontologies. They are used in several components of the system. They are expressed in a representation language based on Frame-Logic (Kifer et al. 95).

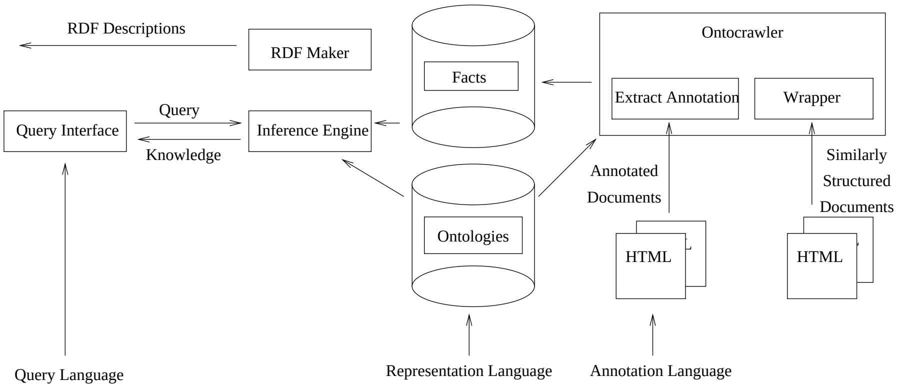


Figure 1. The architecture of ONTOBROKER.

- The Ontocrawler extracts formal knowledge from HTML pages. This is done in two different ways: for large collections of web pages with a similar structure a *wrapper* (Wiederhold&Genesereth 97) generates formal descriptions of the content of the pages in relation to a certain ontology. Often the effort for constructing specialized wrappers is too high: in this case an annotation language is used for enabling providers to enrich web documents with ontological information in an integrated, maintenance-friendly manner.

- The inference engine exploits the formal semantics of the representation language and enables well defined automatic reasoning.

- The RDF-Maker exploits the inference engine and generates an RDF representation of information inferable from the ontology and the facts with respect to a given web resource.

- The query interface enables the interactive formulation of queries while browsing the ontology and selecting the terms constituting the query.

Thus ONTOBROKER is an integrated, comprehensive system to extract, reason and generate domain specific metadata. According to the metadata classification of (Kashyap&Shet 97) our approach deals with *domain-specific metadata* that is *content-descriptive* and utilizes a domain specific ontology. Additionally the metadata we generate is also *direct content-based*, thus allowing semantic-based access to web information. In addition, the reasoning service provides a means for deriving information which has been specified only implicitly in the web sources. The system is fully implemented and can be accessed via (Ontobroker). For a brief introduction of the system cf. (Fensel et al. 98a).

The paper is organized as follows. In section 2, we will present the representation languages and the inference engine used in ONTOBROKER. Section 3 introduces some basics about the Resource Description Framework and the web standards developed by the W3C and relates these developments with the ONTOBROKER approach. We conclude with related work, future work, and a brief summary.

2. The Languages and Inference Engine of ONTOBROKER

In this section we discuss the formalisms used by ONTOBROKER. After describing the representation language used to define ontologies we discuss the query formalism that is used by a client asking for information. Then we present the inference engine that computes the answers to queries. And finally an extension to HTML is presented that allows the smooth integration of ontological annotation in existing web pages.

2.1. THE REPRESENTATION FORMALISM FOR ONTOLOGIES

The basic support we want to provide is answering queries using instances of an ontology. This ontology may be described by taxonomies and rules. Since there are effective and efficient query evaluation procedures for Horn-logic-like languages we based our inference engine on Horn-logic. However, simple Horn-logic is not appropriate from an epistemological point of view for two reasons:

1. The epistemological primitives of simple predicate logic are not rich enough to support adequate representations of ontologies.
2. It is often very artificial to express logical relationships via Horn clauses.

We will subsequently discuss how we overcame both shortcomings.

2.1.1. *Elementary Expressions*

Usually, ontologies are defined via concepts or classes, is-a relationships, attributes, further relationships, and axioms. Therefore an adequate language for defining the ontology has to provide modeling primitives for these notions. Frame-Logic (Kifer et al. 95) provides such modeling primitives and integrates them into a logical framework providing a Horn-logic subset. Furthermore, in contrast to Description Logic, expressing the ontology in Frame-Logic allows queries that directly use parts of the ontology as first class citizens. That is, not only instances and their values but also concept and attribute names can be provided as answers via variable substitutions.

We use a slightly modified variant of Frame-Logic, which suits our needs. Principally the following elementary modeling primitives are used:

- Subclassing: `C1::C2`, meaning that class `C1` is a subclass of `C2`.
- Instance of: `O:C`, meaning that `O` is an instance of class `C`.
- Attribute declaration: `C1[A =>> C2]`, meaning that for the instances of class `C1` an attribute `A` is defined whose value must be an instance of `C2`.
- Attribute value: `O[A->>V]`, meaning that the instance `O` has an attribute `A` with value `V`.
- Part-of: `O1 <: O2`, meaning that `O1` is a part of `O2`.
- Relations: predicate expressions like `p(a1,...,a2)` can be used as in usual logic-based representation formalisms, except that not only terms can be used as arguments but also object expressions.

2.1.2. *Complex Expressions*

From the elementary expressions more complex ones can be built. We distinguish between the following complex expressions: facts, rules, double rules, and queries. Facts are ground elementary expressions. A rule consists of a head, the implication sign `<-`, and the body. The head is just a conjunction of elementary expressions (connected using `AND`). The body is a complex formula built from elementary expressions and the usual predicate logic connectives (implies: `->`, implied by: `<-`, equivalent: `<->`, `AND`, `OR`, and `NOT`). Variables can be introduced in front of the head (with a `FORALL`-quantifier) or anywhere in the body (using `EXISTS` and `FORALL`-quantifiers). A double rule is an expression of the form:

```
head <-> body
```

where the head and body must be conjunctions of elementary expressions. Examples of double rules are given in Table I. An EBNF syntax description of the complete representation language is given in (Fensel et al. 98c).

2.1.3. *An Illustration*

Ontologies defined with this language mainly consist of three parts:

- The concept hierarchy defines the subclass relationship between different classes.

- For classes attribute definitions are given.
- A set of rules defines relationships between different concepts and attributes.

This illustration is taken from the $(KA)^2$-Initiative (Benjamins&Fensel 98) where a community of researchers agrees on an ontology about relevant aspects of a research community. Table I provides part of that ontology. The concept hierarchy consists of elementary expressions declaring subclass relationships. The attribute definitions declare attributes of concepts and the valid types which values of these attributes must have. The first rule ensures symmetry of cooperation and the second rule specifies that whenever a person is known to have a publication, then the publication also has an author who is that particular person and vice versa. This kind of rule completes the knowledge base with information that is distributed and incomplete and thus reduces development as well as maintenance effort. Especially the double rules are very useful, since they explicate e.g. a connection between two object-attribute-value triples. The third rule uses the ontology itself to complete the knowledge base. Based on the schema information missing type information for attribute values are deduced.

2.2. The Query Formalism

The query formalism is oriented towards the syntax of Frame-Logic that defines the notion of instances, classes, attributes, and values. The generic schema for this is:

```
O:C[A->>V]
```

meaning that the object `O` is an instance of the class `C` with an attribute `A` that has a certain value `V`. Variables, constants or arbitrary expressions can be used at each position in the above scheme. Furthermore, because the ontology is part of the knowledge base itself the ontology definitions can be used to validate the knowledge base. In the following we will provide some queries as examples to illustrate our approach.

If we are interested in information about researchers with certain properties. e.g. we want to know the home page, the last name and the email address of all researchers with first name Richard, we achieve this with the following query:

```
FORALL Obj, LN, EM <-
  Obj:Researcher[firstName->>"Richard";
                 lastName->>LN;email->>EM].
```

Table I. A part of an example ontology

Concept Hierarchy	Attribute Definitions

`Object[].`
 	
` Person :: Object.`
 	
`  Employee :: Person.`
 	
`   Researcher :: Employee.`
 	
` Publication :: Object.`
	`Person[`
	` firstname =>> STRING;`
	` lastName =>> STRING;`
	` eMail =>> STRING;`
	` publication =>> Publication;`
	` ...].`
	
	`Employee[`
	` affiliation =>> Organization;`
	` ...].`
	
	`Researcher[`
	` researchInterest =>> Topic;`
	` cooperatesWith =>> Researcher;`
	` ...].`
	
	`Publication[`
	` author =>> Person;`
	` title =>> STRING;`
	` year =>> NUMBER;`
	` abstract =>> STRING].`

Rules

```
FORALL Person1, Person2
  Person1:Researcher[cooperatesWith ->> Person2] <-
    Person2:Researcher[cooperatesWith ->> Person1].

FORALL Person1, Publ1
  Publ1:Publication[author ->> Person1] <->
    Person1:Person[publication ->> Publ1].

FORALL O,C,A,V,T
  V:T <- C[A=>>T] AND O:C[A->>V].
```

In our example ONTOBROKER gives the following answer (actually, there is only one researcher with first name Richard in the knowledge base.

```
Obj = http://www.iiia.csic.es/richard/index.html
LN = Benjamins
EM = mailto:richard@iiia.csic.es
```

Another example asks for the home page of all researchers who cooperate with the researcher with last name Motta:

```
FORALL Obj, CP <-
  Obj:Researcher[lastName ->>"Motta"; cooperatesWith->>CP].
```

The interesting point in this query is that the ontology contains a rule specifying the symmetry of cooperation. That means, even if the researcher with last name Motta did not specify a cooperation with any researcher, ONTOBROKER could deduce such a cooperation, if another researcher stated that he cooperates with Mr. Motta.

Another possibility is to query the knowledge base for information about the ontology itself, e.g. the query

```
FORALL Att, T <- Researcher[Att=>>T]
```

asks for all attributes of the class Researcher and their associated types.

These queries can be posed via a web interface, but since average web users cannot be expected to be familiar with F-Logic a graphical substitution exists that is much more comprehensive. It visualizes the ontology and hides a lot of the unnecessary syntax. A description of this interface can be found in (Fensel et al. 98a).

2.3. Providing Input for Ontobroker

To be able to answer queries, ONTOBROKER needs facts which are stored in its knowledge base. The knowledge base contains knowledge collected from scattered web sources. (Ashish&Knoblock 97) distinguish three classes of web sources:

- *Multiple-instance sources* share the same structure but provide different information, e.g. the CIA World Fact Book (CIA 97), provides information about more than 200 different countries stored on more than 200 similarly structured pages (one page per country).

- *Single-instance sources* provide large amounts of data in a structured format.

- *Loosely structured pages* have no generalizable structure, e.g. personal home pages.

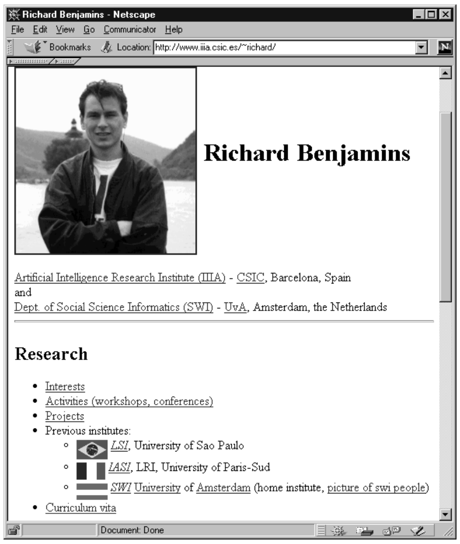
All these sources contain knowledge that should be made accessible by ONTOBROKER. To allow an integration of this knowledge into the knowledge base it has to be formalized. This can be done in two ways:

Sources falling into the first two categories allow us to implement wrappers (Wiederhold&Genesereth 97) that automatically extract factual knowledge from these sources. If the structure of the pages is known and stable over time these wrappers can automatically create parts of the knowledge base of ONTOBROKER and thus allow inferencing and query answering about the provided information. We applied this approach to the CIA World Fact Book using a simple ontology about countries and their characteristics.

The second way to provide a formal representation of unstructured information is based on manual work. Since formalization in the third case mentioned above can hardly be achieved automatically we chose a manual annotation approach to capture loosely structured information. Large amounts of the information provided in the WWW are formulated using the Hyper-Text Mark-up Language (HTML) on hardly structured pages. We developed a minor extension to the HTML syntax (the onto-attribute) to enable ontological annotation of web pages. Annotating resources with semantic information has certain advantages over simple meta-tagging of resources, i.e.:

- The embedded annotations are located physically close to the rendered information they belong to.
- The semantic information is in part represented as the informal text of the resource, i.e. the text can be reused in a formal way, e.g. as the value of attributes.
- In the same way hyper links contained on web pages can be reused to establish formal relations between concepts.

The general idea behind our approach (see (Fensel et al. 98c) for more details) is to take an HTML page as a starting point and to add only few ontologically relevant tags to its mark-up. By these minor changes the information contained in the page is annotated and made accessible as facts to ONTOBROKER. This approach allows providers to annotate their web pages gradually, i.e. they do not have to completely formalize the knowledge contained therein. Further, the pages remain readable by standard browsers. Thus, there is no need to keep several different sources up-to-date and consistent which reduces development as well as maintenance effort considerably. All factual ontological information is contained in the HTML mark-up itself.




```
<HTML><HEAD>
<a onto="page:Researcher"> </a>
<TITLE> Richard Benjamins </TITLE>
</HEAD>
<H1> <A HREF="pictures/id-rich.gif">
<IMG SRC="pictures/richard.gif"></A>
<a onto="page[photo=href]"
   HREF="http://www.iiia.csic.es/richard/
            pictures/richard.gif" ></a>

<a onto="page[firstName=body]">Richard</a>
<a onto="page[lastName=body]">Benjamins</a>
</h1> <p>

<A onto="page[affiliation=body]">
Artificial Intelligence Research Institute
(IIIA)</A> -
<a href="http://www.csic.es/">CSIC</a>,
Barcelona, Spain <br>
```

Figure 2. An example of an annotated web page.

We provide three different epistemological primitives to annotate ontological information in web documents:

1. An object identified by a URL (Uniform Resource Locator) can be defined as an instance of a certain class.

2. The value of an object's attribute can be set.

3. A relationship between two or more objects may be established.

All three primitives are expressed by using an extended version of a frequent HTML tag, i.e. the anchor tag.

Typically a provider of information first defines an object. This is done by stating which class of the ontology it is an instance of. For example, if Richard Benjamins (his home page and a part of its sources are depicted in Figure 2) would like to define himself as a researcher, he would say the URL of his home page is an instance of the class researcher. To express this in our HTML extension he uses the following line on his home page.

```
<a onto=" 'http://www.iiia.csic.es/richard' : Researcher">
```

The identifier `'http://www.iiia.csic.es/richard'` denotes an object, namely an instance of class researcher. Actually this id is the URL of Richard Benjamins' home page, thus, from now on he as a researcher is denoted by the URL of his home page (see Figure 2).

Each class is associated with a set of attributes. Each instance of a class can define values for these attributes. To define an attribute value on a web page the knowledge provider has to list the object, the attribute, and the value. For example, the ontology contains an attribute email for each object of class researcher. If Richard Benjamins wants to provide his email address, he uses this line on his home page.

```
<a onto=" 'http://www.iiia.csic.es/richard'
             [email='mailto:richard@iiia.csic.es'] ">
```

This line states that the object denoted by the handle has the value `'mailto:richard@iiia.csic.es'` for the attribute email.

Several objects and attributes can be defined on a single web page, and several objects can be related to each other explicitly. Given the name of a relation `REL` and the object handles `Obj1` to `Objn` this definition looks like this:

```
<a onto= "REL(Obj1, Obj2, Obj3, ..., Objn)" >
```

The listed examples look rather clumsy, esp. because of their long object handles and the redundancy due to writing information twice, once for the browser and a second time for ONTOBROKER. So the annotation language provides some means to ease annotating web pages and get rid of a big share of the clumsiness and redundancy (Fensel et al. 98c). A set of keywords with special meanings is allowed as part of the annotation syntax. The keyword *page* represents the whole web page where the ontological mark-up is contained. This is useful when looking at the page as a representative of an object. For example, a home page of a researcher might represent that person in the knowledge base. This can be defined by the following kind of annotation:

```
<a onto= "page:Researcher">
```

The following annotation defines the affiliation attribute of the object denoted by the URL of the current page and takes the value from the anchor-tag's href-attribute.

```
<a onto="page[affiliation=href]"
   href="http://www.iiia.csic.es/">
```

The *href* keyword allows us to establish relations between objects without a lot of typing, because the hyper-links can be reused within the ontological mark-up.

Not only hyper-links can be directly integrated as semantic information, the text that is rendered by a browser can also become a part of the formal knowledge, e.g.

Table II. Principle mechanism for translating F-Logic to predicate logic

Frame Logic	Meaning	Predicate Logic
`C1 :: C2`	class `C1` is a subclass of `C2`	$sub(C1, C2)$
`O : C`	`O` is an instance of class `C`	$isa(O, C)$
`C1[A=>>C2]`	for the instances of `C1` an attribute `A` is defined, whose value must be an instance of `C2`	$att\_type(C1, A, C2)$
`O[A->>V]`	the instance `O` has an attribute `A`, whose value is `V`	$att\_val(O, A, V)$
`O1 <: O2`	`O1` is a part of `O2`	$part\_of(O1, O2)$

```
<a onto="page[firstName=body]> Richard </a>
```

defines Richard (contained between `<a ...>` and `</a>`) as an attribute value for firstName. The keyword *body* allows this kind of reuse. Through these conventions the annotation of web pages becomes more concise and redundancy can be nearly avoided. This tight coupling eases metadata maintenance for frequently changing resources, since changing the rendered data is automatically reflected in the semantic mark-up.

Although the technique just presented is currently tailored towards HTML, it can be easily adapted for any XML based mark-up language: the only changes required are slight modifications of the respective document type definition (DTD) of that language. This is especially important since more and more applications of XML languages are currently developed.

2.4. The Inference Engine of Ontobroker

The inference engine of Ontobroker has two key parts: the one that does the translation (and retranslation) process from the rich modeling language (F-Logic) to a restricted one (Horn logic) and the part that does the evaluation of expressions in the restricted language.

The input of the inference engine consists of the ontology, collected facts from the web and queries formulated in Frame-Logic. We have decided against direct evaluation of expressions of the rich modeling language. There are techniques known for evaluating Frame-Logic (Frohn et al. 97), but they do not support the whole language and the semantics we need (e.g. full first order rule bodies). Furthermore a direct evaluation approach would be very inflexible, a small change in the input language would result in changes of the whole system and building a specialized inference engine for a special semantics requires an extraordinary effort. Instead a Frame-Logic-translator translates the Frame-Logic expressions via several intermediate states to first-order logic expressions. Table II gives an idea of how this translation is performed. After several transformation steps (cf. (Decker 98), (Fensel et al. 98c) for more details) we obtain a normal logic program. Techniques from deductive databases are applicable to implement the bottom-up fix-point evaluation procedure. Because we allow negation in the clause body we have to carefully select an appropriate semantics and evaluation procedure. If the resulting program is stratified, we use simple stratified semantics and evaluate it with a technique called dynamic filtering (Kifer&Lozinskii 86) (Fensel et al. 98d). But the translation of Frame Logic usually results in a logic program with only a limited number of predicates (all object expressions are compiled into the same predicate), so the resulting program is often not stratified. To deal with non-stratified negation we have adopted the well-founded model semantics (VanGelder et al. 91) and compute this semantics with an extension of dynamic filtering.

3. Web Standards and Ontobroker

3.1. RDF/RDFS and Frame-Logic

In the WWW the need for a standardized notation for metadata led to the development of the Resource Description Framework (RDF) by the W3C. RDF is a framework for describing general-purpose metadata that is richer than simple keyword based metadata annotations, since it introduces the notion of resources. Resources are objects that can have certain properties and can be related to other resources (cf. (RDF) for the current status of the framework definition). Any object that can be addressed via a URL may be a resource in the sense of RDF. Since a resource together with attached properties and values can be used again as a resource, this representation style allows us to build labeled directed graphs that resemble semantic nets.

A proposed syntax for RDF uses XML so that RDF specifications can be easily integrated in applications following the current trend towards XML as *the* language for sharing information. Due to that RDF will probably become a widely recognized language and representation formalism for metadata that can serve as an interlingua for information interchange.

RDF is complemented by a schema definition language (RDF Schema) (RDFS). RDFS is a format for defining the terminology that can be used to describe RDF data. It basically allows us to define classes, attributes (property types), value ranges and cardinality constraints for property types. `RDF:instanceOf` and `RDFS:subClassOf` are examples of predefined property types, which correspond to similar notions in frame-based or object-oriented languages. So RDFS allows the definition of ontologies for RDF specifications in a way which has some similarities to F-Logic-based ontologies.

However there exist some major differences:

- Both representation formalisms support an (object, attribute, value) view on the object level, and a (class, attribute, type) view on the schema level, so both have a similar kind of representation.

- F-Logic supports inference rules which can be used to make implicit knowledge explicit, e.g. to derive attribute values of objects.

- F-Logic has a well defined semantics and proof theory, thus building an inference engine for it is a clearly defined task, whereas the semantics of RDF still has to be defined formally.

- RDF supports the reification of resource descriptions, i.e. an RDF expression (consisting of a resource, a property type, and a value) can be the resource of another description. This is not possible in F-Logic.

- The schemas of RDF allow the definition of attributes, so called property types. These property types are —in contrast to frame based languages like F-Logic— general in the sense that they do exist independently of classes. Thus, it is not possible to give the same name to different properties for several classes if they have different value ranges or cardinalities.

3.2. What has Ontobroker to offer to RDF?

RDFMaker The kinds of information that can be stored in RDF metadata include concepts that are stored in the ontological annotations for Ontobroker. To make this information accessible to a wider community we developed a tool (RDFMaker, cf. (Erdmann et al. 98) and figure 1) that translates these annotations (in Ontobroker syntax) to metadata (in RDF syntax). The tool takes an annotated web page and computes all inferable information based on the ontology and the annotated facts. Subsequently, it formulates all derived information according to the RDF definition and adds it to the source. In this way any information seeker being capable of understanding RDF (e.g.

information agents) can profit from the annotation made for ONTOBROKER. Thus the advantages of ontological annotations of resources and the homogeneity, accessibility and wide dissemination (at least in the future) of RDF metadata descriptions are combined.

Maintenance and Redundancy Reduction RDF defines a portable way of expressing metadata, but it is separated from the data. So maintenance of metadata might result in high effort: if the data change, the metadata also has to be changed to keep both in sync. A better approach is to combine both aspects. In ONTOBROKER we use annotations that are included inside the data and directly refer to the information contained in the pages, thus ontological information can be automatically extracted and therefore is always consistent and up-to-date. When using RDFMaker to automatically generate RDF descriptions from the ONTOBROKER annotations, the problem of maintaining metadata can be reduced. At the same time the degree of redundancy is lowered because information from the HTML pages is directly incorporated in the metadata by RDFMaker.

Inferencing Although RDF/RDFS does not allow the formulation of rules, there exist useful inference tasks for RDF. The property type `RDFS:subClassOf` is transitive (RDFS, section 2.2.2), thus information seekers looking for all instances of a special class `C` should retrieve all instances of all subclasses of `c` as well. Another example for a useful inference task is the deduction of implicit information. RDFS allows to restrict the ranges of property types. This information could be used to infer `RDFS:instanceOf` relations and thus explicating implicit information. For example, if the property type `cooperatesWith` has the range restriction researcher, any resource that is the value of this property type can be inferred as belonging to class researcher. This is desirable, because knowledge on the WWW is often incomplete and this is a possibility to make it more complete.

Nevertheless, there is (as far as we know) no system available that contains an inference mechanism for RDF. To be able to handle inference tasks and —more general— rules we propose to use RDF as a representation language for metadata and F-Logic as the basis for the inference engine. Thus, RDF/RDFS should be used to represent metadata within the websources and F-Logic should be used when answering queries that are based on an ontology (including rules). This combination of a generally accepted and standardized representation language and a powerful and flexible inference engine would drastically enhance the power and usability of RDF. The ONTOBROKER-system has already proved the feasibility of this combination.

4. Conclusions, Related and Future Work

Up to now, the inference capabilities of the WWW are very limited. In essence, they are restricted to keyword-based search facilities which are offered by the various web services. This is clearly not sufficient when dealing with reusable multimedia data on the WWW. As a way to overcome these problems ontologies and metadata were proposed by several authors (Fensel et al. 98a) (Kashyap&Shet 97) (Mena et al. 98) (Boll et al. 98) and led to a number of systems.

Similar approaches to ours in regard to metadata are InfoHarness (Shklar et al. 95) and Observer (Mena et al. 96). InfoHarness extracts metadata with a kind of wrappers. Information brokering is done primarily on the level of representation and not based on domain specific ontologies. E.g. mainly metadata like author, title, file size etc. are extracted and used for query answering. Therefore, large ontologies with rules are not supported; inferences are not possible.

The Observer system can be seen as a successor of InfoHarness: it aims at integrating multiple information sources, each with its own domain specific ontology. A user poses a query in his own user ontology. This query is translated using synonyms to queries according to the component ontology and evaluated by the component systems. Observer focuses on integrating multiple ontologies, and thus several aspects are different from ONTOBROKER. In ONTOBROKER it is possible to specify rules that express dependencies between different terms from the ontology and to complete information using the ontology itself. Because Observer uses description logics this is not possible in Observer. Furthermore, ONTOBROKER is a complete approach supporting a user with an annotation language, an inference engine and a graphical query interface, while support like this is not available for the Observer system.

Another approach similar to ours is SHOE (Luke et al. 97) which introduced the idea of using ontologies to annotate information in the WWW. HTML pages are annotated via ontologies to support information retrieval based on semantic information. However, there are major differences in the underlying philosophy: In SHOE, providers of information can introduce arbitrary extensions to a given ontology. Furthermore, no central provider index is defined. As a consequence, when specifying a query the client may not know all the ontological terms which have been used to annotate the HTML pages and the web crawler has to visit the entire WWW to ensure to find all annotated knowledge fragments. The answers given to a query may be incomplete because the used ontologies are not entirely known and the web crawler cannot find all relevant pages.

In contrast, ONTOBROKER relies on the notion of an *Ontogroup* and domain specific ontology defining a group of web users that agree on an ontology for a given subject. Therefore, both the information providers and the clients have complete knowledge of the available ontological terms. In addition, the ontogroup is stored in a provider index used by Ontocrawler when collecting all annotated HTML pages. Thus, ONTOBROKER can deliver complete answers to the posed queries. The philosophy of ONTOBROKER is also tailored to homogeneous intranet applications, e.g. for knowledge management within an enterprise. In this context the information providers are well known and the ontology can be fixed because in the enterprise a common view on the world should exist.

SHOE and ONTOBROKER also differ with respect to their inferencing capabilities. SHOE uses description logic as its basic formalism, currently offers rather limited inferencing capabilities and does not support RDF. ONTOBROKER relies on Frame-Logic and supports more complex inferencing for answering queries (see (Kandzia&Schlepphorst 97) (Fensel et al. 98b) for a comparison of the two representation and reasoning paradigms).

Because ontologies and metadata are means to overcome the restriction of the current capabilities to access the web the definition, representation, extraction and maintenance of metadata are questions that have to be solved. This paper presented ONTOBROKER, a system that addresses these tasks. ONTOBROKER uses F-Logic to define the ontology and to represent a knowledge base that allows inferencing. Metadata extraction from a web page is done either by wrappers or by a web crawler that identifies special semantic tagging in web pages. In ONTOBROKER this annotation information is tightly integrated into the HTML mark-up. This reduces redundancy of information and makes maintenance of metadata a simpler task since metadata can easily be generated (e.g. in RDF) when changes in the original sources occur. The techniques developed for annotations are transferable to all XML-based languages.

ONTOBROKER provides means for semantic-based query handling even if the information is spread over several sources. Furthermore, the generation of RDF descriptions enables the exploitation of the ontological information in RDF-based applications —intelligent agents can use the knowledge provided by the RDF descriptions. The system is currently the basis for realizing the Knowledge Acquisition Initiative $(KA)^2$ (Benjamins&Fensel 98) (Benjamins et al. 98) and for developing a knowledge management system for industrial designers in regard to ergonomic questions. In the latter project, the same knowledge may be used by humans and for inferences of the system. This twofold use of the same piece of knowledge is enabled through the tight coupling of semi-formal and formal knowledge in ONTOBROKER.

Acknowledgements

We thank V. R. Benjamins, A. Gomez-Perez and R. Perkuhn for their helpful comments. Special thanks to J. Angele who developed the evaluation procedure for L-KARL that is used by ONTOBROKER. The CIA World Fact Book wrapper for ONTOBROKER was developed by A. Dagasan and A. Witt.

References

[Ashish&Knoblock 97] N. Ashish and C. Knoblock: Semi-automatic Wrapper Generation for Internet Information Sources. In Proceedings of the IFCIS Conference on Cooperative Information Systems (CoopIS), Charlston, South Carolina 1997.

[Benjamins et al. 98] V. R. Benjamins, D. Fensel and A. Gomez Perez: Knowledge Management Through Ontologies. In: *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management (PAKM'98)*, Basel, Switzerland, October 1998.

[Benjamins&Fensel 98] V. R. Benjamins and D. Fensel: The Ontological Engineering Initiative (KA)2. In N. Guarino (Ed.), *Formal Ontologies in Information Systems, Frontiers in Artificial Intelligence and Applications*, IOS-Press, Amsterdam, 287-301, 1998.

[Boll et al. 98] S. Boll, W. Klas and A. Sheth: Overview on Using Metadata to Manage Multimedia Data. In: (Sheth&Klas 98), pp. 1-24, 1998

[CIA 97] CIA World Fact Book 1997, http://www.odci.gov/cia/publications/factbook

[Decker 98] S. Decker. On Domain-Specific Declarative Knowledge Representation and Database Languages In: *Proceedings of the 5th KRDB Workshop (KRDB98)*, Seattle, WA, 31-May-1998, Eds: A. Borgida, V. Chaudri, M. Staudt

[Erdmann et al. 98] M. Erdmann, S. Decker, D. Fensel, and R. Studer: Combining ONTOBROKER with the Standards of the WWW. research report, Institute AIFB, University of Karlsruhe, 1998.

[Etzioni 97] O. Etzioni: Moving Up the Information Food Chain, *AI Magazine*, 18(2), 1997.

[Farquhar et al. 97] A. Farquhar, R. Fikes, and J. Rice: The Ontolingua Server: a Tool for Collaborative Ontology Construction, *International Journal of Human-Computer Studies (IJHCS)*, 46(6):707728, 1997.

[Fensel et al. 98a] D. Fensel, S. Decker,M. Erdmann, and R. Studer: ONTOBROKER: The Very High Idea. In: *11th Florida Artificial Intelligence Research Symposium (FLAIRS-98)*, Sanibal Island, Florida, May 1998

[Fensel et al. 98b] D. Fensel, M.-C. Rousset, and S. Decker: Workshop on Comparing Description and Frame Logics, *Data and Knowledge Engineering* 25(3):347-352, 1998.

[Fensel et al. 98c] D. Fensel, S. Decker, M. Erdmann, and R. Studer: ONTOBROKER: How to make the WWW Intelligent, research report no. 376, Institute AIFB, University of Karlsruhe, 1998. http://www.aifb.uni-karlsruhe.de/WBS/broker.

[Fensel et al. 98d] D. Fensel, J. Angele, and R. Studer: The Knowledge Acquisition and Representation Language, KARL. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10, No. 4, 1998.

[Fridman Noy&Hafner 97] N. Fridman Noy and C. D. Hafner: The State of the Art in Ontology Design, *AI Magazine*, 18(3):53-74, 1997.

[Frohn et al. 97] J. Frohn, R. Himmeröer, P.-Th. Kandzia, G. Lausen, and C. Schlepphorst: FLORID - A Prototype for F-Logic, In: *Proceedings of the International Conference on Data Engineering (ICDE, Exhibition Program)*, Birmingham, 1997.

[Gruber 93] T. R. Gruber: A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, 5(2), 1993.

[Hammer et al. 97] J. Hammer, H. Garcia-Molina, J. Cho, A. Crespo, R. Aranha: Extracting Semistructured Information from the Web. In: *Proceedings of the Workshop on Management of Semistructured Data*, pages 18-25, Tucson, Arizona, May 1997.

[Kandzia&Schlepphorst 97] P.-T. Kandzia and C. Schlepphorst: DOOD and DL - Do We Need an Integration. In: *Proceedings of the 4th KRDB Workshop*, Athens, Greece, August 30, 1997.

[Kashyap&Shet 97] V. Kashyap and A. Sheth: Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies. In: M. Papazoglou and G. Schlageter (Eds.): *Cooperative Information Systems: Current Trends and Directions*, Academic Press, 1997.

[Kifer et al. 95] M. Kifer, G. Lausen, and J. Wu: Logical Foundations of Object-Oriented and Frame-Based Languages, *Journal of the ACM*, 42, 1995.

[Kifer&Lozinskii 86] M. Kifer, E. Lozinskii: A Framework for an Efficient Implementation of Deductive Databases. In: *Proceedings of the 6th Advanced Database Symposium*, Tokyo, 1986.

[Klein&Frankhauser 97] B. Klein and P. Fankhauser: Error tolerant document structure analysis. *International Journal on Digital Libraries* 1:344-257, Springer, 1997

[Lamping et al. 95] L. Lamping, R. Rao, and Peter 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*, 1995

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

[Mena et al. 96] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi: OBSERVER: An approach for query processing in global information systems based on interoperation across preexisting ontologies. In: *Proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, June 1996

[Mena et al. 98] E. Mena, V. Kashyap, A. Illarramendi, and A. Sheth: Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure, In: *Intl. Conf. on Formal Ontology in Information Systems (FOIS'98)*, Treno, Italy, June 1998.

[Ontobroker] ONTOBROKER: http://www.aifb.uni-karlsruhe.de/WBS/broker

[PGML] Precision Graphics Markup Language (PGML), World Wide Web Consortium Note 10-April-1998, http://www.w3.org/TR/1998/NOTE-PGML

[RDF] Resource Description Framework (RDF), W3C Working Draft 19 August 1998, http://www.w3.org/ TR/1998/WD-rdf-syntax-19980819

[RDFS] Resource Description Framework Schema (RDFS), W3C Working Draft 14 August 1998, http:// www.w3.org/TR/1998/WD-rdf-schema-19980814

[Schlepphorst 97] C. Schlepphorst: Semi-naive Evaluation of F-Logic Programs, Technical Report 85 , University of Freiburg, 1997

[Sheth&Klas 98] A. Sheth and W. Klas (Eds.): *Multimedia Data Management: Using Metadata to Integrate and Apply Digital Media*, McGraw-Hill, March 1998

[Shklar et al. 95] L. Shklar, A. Sheth, V. Kashyap, and K. Shah: InfoHarness: Use of Automatically Generated Metadata for Search and Retrieval of Heterogeneous Information. In: *Proceedings of CAiSE '95*, Jyvaskla, Finland, June 1995, Lecture Notes in Computer Science 932, Springer.

[SMIL 1.0] Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, W3C Recommendation 15-June-1998, http://www.w3.org/TR/1998/REC-smil-19980615/

[VanGelder et al. 91] A. Van Gelder, K. Ross, J. S. Schlipf: The Well-Founded Semantics for General Logic Programs, *Journal of the ACM*, 38(3): 620650, 1991.

[Wiederhold 92] G. Wiederhold: Mediators in the Architecture of Future Information Systems, *IEEE Computer*, 25(3):3849, 1992.

[Wiederhold&Genesereth 97] G. Wiederhold and M. Genesereth: The Conceptual Basis for Mediation Services. *IEEE Expert*, September/October, pp. 38-47,1997.

[XML] Extensible Markup Language (XML), http://www.w3.org/TR/PR-xml-971208.