!Remark to reviewers: The manuscript is currently also under review of a native speaker to clean the language!

Lessons Learned from Applying AI to the Web

Dieter Fensel, Jürgen Angele, Stefan Decker, Michael Erdmann, Hans-Peter Schnurr, Rudi Studer and Andreas Witt

Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
dfe@aifb.uni-karlsruhe.de, http://www.aifb.uni-karlsruhe.de/~dfe

Abstract. Ontobroker applies Artificial Intelligence techniques to improve access to heterogeneous, distributed and semistructured information sources as they are presented in the World Wide Web or organization-wide intranets. It relies on the use of *ontologies* to annotate web pages, formulate queries and derive answers. In the paper we will briefly sketch Ontobroker. Then we will discuss its main shortcomings, i.e. we will share the lessons we learned from our exercise. We will also show how On2broker overcomes these limitations. Most important is the separation of the query and inference engines and the integration of new web standards like XML and RDF.

1 Introduction

The World Wide Web (WWW) currently contains around 300 million static objects providing a broad variety of information sources (cf. [Bharat & Broder, 1998]). The early question of whether a certain piece of information is on the web has become the problem of how to find and extract it. The problem will become even more serious if the web continues to grow at the high speed expected by the W3C (the standardization committee of the WWW). Therefore, dealing with the problem of finding and accessing information in the WWW has become a key issue in overcoming the information overload, i.e. in preventing users from wasting hours in going through useless information and trying to find the piece of information they are interested in.

Artificial Intelligence (AI) has a strong tradition of developing methods, tools and languages for structuring knowledge and information. Therefore it is quite natural to apply its techniques to tackle the above problems. Viewing the web as a large knowledge-based system, however, makes aware of its very limited querying and inference interface at its current state. In the area of knowledge-based systems *ontologies* have been developed for structuring and reusing large bodies of knowledge (cf. CYC [Lenat, 1995], KIF/Ontolingua [KIF], and CommonKADS [Schreiber et al., 1994]). Ontologies are consensual and formal specification of a vocabulary used to describe a specific domain. Frame-based languages enriched by logical axioms are often used to formulate them (cf. LOOM [MacGregor, 1990] and Frame Logic [Kifer et al., 1995]). Roughly, ontologies correspond to generalized database schemata. However, ontologies can be used to describe the semantic structure of much more complex objects than common databases and are therefore well-suited for describing heterogeneous, distributed and semistructured information sources. The way ontologies are used in On2broker may rather be compared with mediating schemes as used in multi-database systems. There mediating schemes are used to provide an integrated view on the different local database schemes. However, these schema description languages are typically less expressive than our ontology specification language, especially with respect to the rule language part.[1] In the meantime a number of projects rely on such notions (cf. HERMES [Subrahmanian, to appear], Infomaster [Genesareth et al., 1997], and Information Manifold [Levy et al., 1996]) for integrating information sources. SHOE (cf. [Luke et al., 1996], [Luke et al., 1997]) and Ontobroker (cf. [Fensel et al., 1998a], [Decker et al., 1999]) use ontologies for information mediation focussing on the integration of HTML sources distributed throughout the WWW.

Ontobroker provides a broker architecture with three core elements: a query interface for formulating queries, an inference engine used to derive answers, and a webcrawler used to collect the required knowledge from the web. It provides a *representation* language for formulating ontologies. A subset of it is used to formulate queries, i.e. to define the *query language*. An *annotation* language is offered to enable knowledge providers to enrich web documents with ontological information. The strength of Ontobroker is the tight coupling of informal, semiformal and formal information and knowledge. This supports their maintenance and provides a service that can be used more generally for the purpose of *knowledge management* and for integrating knowledge-based reasoning and the semiformal representation of documents.

Applying these techniques to the web and to scenarios of realistic size, however, creates a couple of serious problems and brings up some interesting new insights. We will address the most interesting ones in this paper. Some of the above discussed problems and learned lessons could directly be addressed by Ontobroker and some of them required the redesign of the system now called On2broker. The major new design decisions in On2broker are the clear separation of query and inference engines and the integration of new web standards like XML and RDF. Both decisions are answers to two significant complexity problems of Ontobroker: the computational inference effort for a large number of facts and the human annotation effort for adding semantics to HTML documents.[2]

The content of the paper is organized as follows. First, we describe Ontobroker in section 2 and show how it enables integrated access to HTML pages distributed throughout the WWW. Then, we discuss in section 3 the limitations of this approach. These limitations illustrate the difficulties when making a step forward from solving toy examples to a

---

[1] Also, ontologies have a much broader application context than database schemes. They are means to formalize joint domain theories and not necessarily reflects the conceptual or logical structure of a data store.

[2] In terms of the database community On2broker is a kind of data warehouse for data on the Web. Queries are not run on the sources themselves to which On2broker provides access, but on a database into which the source content has been extracted. In addition to the facts that can be found explicitly in the sources, the system applies also rules to derive additional information.

system that is useful for solving real-world problems. Section 4 draws the consequences and introduces On2broker which overcomes most of the serious limitations of its predecessor. We will also show that such a system can successfully handle a much broader scope of tasks than intended at its inception. Section 5 provides the scope of tasks and domains, On2broker can be applied to. Our conclusions, and a discussion of related and future work are provided in section 6.

2 Ontobroker and its Merits

Ontobroker uses ontologies for information mediation focussing on the integration of HTML sources distributed over the WWW. To achieve this goal, Ontobroker provides three interleaved languages and two tools.

2.1 The Languages

Ontobroker provides an *annotation* language called HTML^A to enable the annotation of HTML documents with machine-processable semantics. For example, the following HTML page states that the text string „Richard Benjamins“ is the name of a researcher where the URL of his homepage is used as his object id.

> <html><body><a *onto="page:Researcher"*><h2>Welcome to my homapge</h2>
> My name is <a *onto="[name=body]"*>Richard Benjamins.</body></html>

An important design decision of HTML^A was

(1) to smoothly integrate semantic annotations into HTML and

(2) to prevent the duplication of information.

The reason for the former decision was to lower the threshold for using our annotation language. People who are able to write HTML can use it straightforwardly as a simple extension. The rationale underlying the second decision is more fundamental in nature. We do not want to add additional data, *instead we want to make explicit the semantics of already available data*. The same piece of data (i.e., „Richards Benjamins“) that is rendered by a browser is given a semantic in saying that this ascii string provides the name of a researcher. We will later show that this is a significant difference between our approach and approaches like SHOE, RDF, and annotations used in information retrieval.

In terms of a knowledge-based system, the annotation language provides the means to express factual knowledge (ground literals). Further knowledge in are provided by the ontology. The ontology defines the terminology (i.e., signature) and may introduce further rules (i.e., axioms) that allow the derivation of additional facts that are not stated extensionally.

A *representation* language is used to formulate an ontology. This language is based on Frame logic [Kifer et al., 1995]. Basically it provides classes, attributes with domain and range definitions, is-a hierarchies with set inclusion of subclasses and multiple attribute inheritance, and logical axioms that can be used to further characterize relationships

```
Object[].
  Person :: Object.
  Publication::Object.
Person[
   firstName =>> STRING;
   lastName =>> STRING;
   eMail =>> STRING;
   ...
   publication =>> Publication].
Publication[
author =>> Person;
title =>> STRING;
year =>> NUMBER;
abstract =>> STRING].
```

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

FORALL Person1, Publication1
   Publication1:Publication [author ->> Person1]
<->
  Person1:Person [publication ->> Publication1].
```

c_1 :: c_2 means that c_1 is a subclass of c_2.
$c[a$ ==> $r]$ means that an attribute a is of domain c and range r.
$o : c[a$->> $v]$ means that o is element of c hand has the value v for a.
<- means logical implication and <-> logical equivalence.

Figure 1 An excerpt of an ontology (taken from [Benjamins et al., to appear])

between elements of an ontology and its instances. The representation language introduces the terminology that is used by the annotation language to define the factual knowledge provided by HTML pages on the web. A little example is provided in Figure 1. It defines the class *Object* and its subclasses *Person* and *Publication*. There some attributes are defined and some rules expressing relationships between them, for example, if a publication has a person as an author then the author should have it as a publication. Semantically, the language for defining rules is the fragment of first-order logic that can be transformed via Lloyd-Topor transformations [Lloyd & Topor, 1984] into Horn logic. Syntactically it differ as it incorporates object-oriented modeling primitives.

The *query* language is defined as a subset of the representation language. The elementary expression is:

$$x \in c \wedge attribute(x) = v$$

written in Frame logic:

$$x[attribute \text{ -> } v] : c$$

In the head of F-Logic rules variables are all quantified. In the body variables may be either all or existentially quantified. All quantified variables must additionally be bound by a positive Atom in the body. Lloyd-Topor transformation handles these quantifications as following. Existential quantifiers in the body may be dropped, because in the body of a rule every variable is implicitly existentially quantified. An all quantification, *forall* y $p(y)$, in the body is transformed to a $\neg$ *exists* y $\neg$ $p(y)$. Then LLoyd-Topor transformation produces a set of rules out of this. Queries are handled as rules without head. Thus the above mentioned conditions for quantifications hold here too.

Complex expressions can be built by combing these elementary expressions with the usual logical connectives ($\wedge$, $\vee$, $\neg$). The following query asks for all abstracts of the publications of the researcher „Richard Benjamins“.

x[*name* -> „Richard Benjamins“; *publication* ->> { *y*[*abstract* -> *z*]}] : *Researcher*

The variable substitutions for *z* are the desired abstracts.

2.2 The Tools

Ontobroker relies on two tools that give it „life“: a *webcrawler* and an *inference engine*. The *webcrawler* collects web pages from the web, extracts their annotations, and parses them into the internal format of Ontobroker. The *inference engine* takes these facts together with the terminology and axioms of the ontology, and then derives the answers to user queries. To achieve this it has to do a rather complex job. First it translates Frame logic into Predicate logic and second it translates Predicate logic into Horn logic via Lloyd-Topor transformations [Lloyd & Topor, 1984]. The translation process is summarized in Figure 2

As a result we obtain a normal logic program. Standard techniques from deductive databases are applicable to implement the last stage: the bottom-up fixpoint 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 (cf. [Kifer & Lozinskii, 1986], [Angele, 1993]) which focuses the inference engine to the relevant parts of a minimal model required to answer the query. Dynamic filtering combines bottom-up and top-down evaluation techniques. The top-down part restricts the set of facts which has to be computed to a subset of the minimal model only. *Thus also infinite minimal models are possible because only this subset has to be finite* [Fensel et

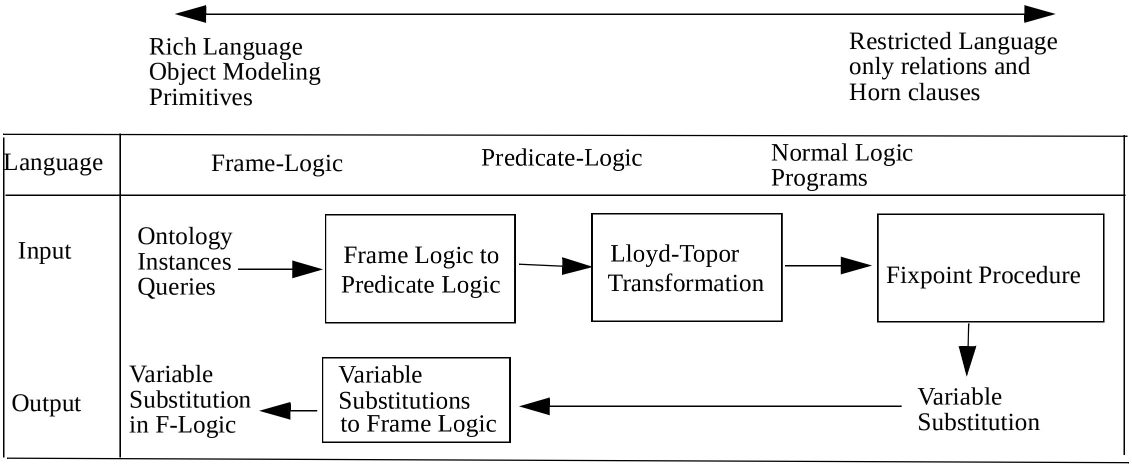
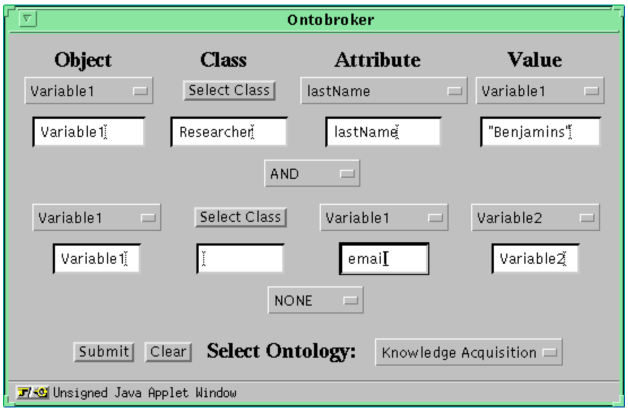


Figure 2 Stages and Languages used in the Inference Engine



Object	Class	Attribute	Value
Variable1	Select Class	lastName	Variable1
Variable1	Researcher	lastName	"Benjamins"

AND

Object	Class	Attribute	Value
Variable1	Select Class	Variable1	Variable2
Variable1		email	Variable2

NONE

Select Ontology: Knowledge Acquisition

Figure 3 The query interface.

al., 1998b].[3] The translation of Frame Logic usually results in a logic program with only a limited number of predicates, so the resulting program is often not stratified. In order to deal with non stratified negation we have adopted the *well-founded model semantics* [Van Gelder et al., 1991] and compute this semantics with an extension of dynamic filtering.

We developed a hyperbolic presentation of the ontology and a tabular interface to improve accessibility of the inference engine. Expecting a normal web user to type queries in a logical language and to browse large formal definitions of ontologies is not very practical. Therefore, we exploited the structure of the query language to provide a tabular query interface as shown in Figure 3. There we ask for the researchers whose last name is *Benjamins* and their email addresses. We also need support for selecting classes and attributes from the ontology. To allow the selection of classes, the ontology has to be presented in an appropriate manner. Usually an ontology can be represented as a large hierarchy of concepts. With regards to the handling of this hierarchy a user has at least two requirements: first he wants to scan the vicinity of a certain class looking for classes better suitable to formulate a certain query. Second a user needs an overview of the entire hierarchy to allow for a quick and easy navigation from one class in the hierarchy to another class. These requirements are met by a presentation scheme based on Hyperbolic Geometry [Lamping et al., 1995] where classes in the center are depicted with a large circle, whereas classes at the border of the surrounding circle are only marked with a small circle (see Figure 4). The visualization technique allows a quick navigation to classes far away from the center as well as a closer examination of classes and their vicinity. When a user selects a class from the hyperbolic ontology view, the class name appears in the class field of the tabular interface and the user can select one of the attributes from the attribute choice menu as the pre-selected class determines the possible attributes. Based on these interfaces Ontobroker automatically derives the query in textual form and presents the result of the query.

Ontobroker[4] is available on the web and has been applied in a few applications in the

---

[3] Syntactical rules that ensure that the subset of minimal model that has to be computed remains finite are described in [Fensel et al., 1998b].

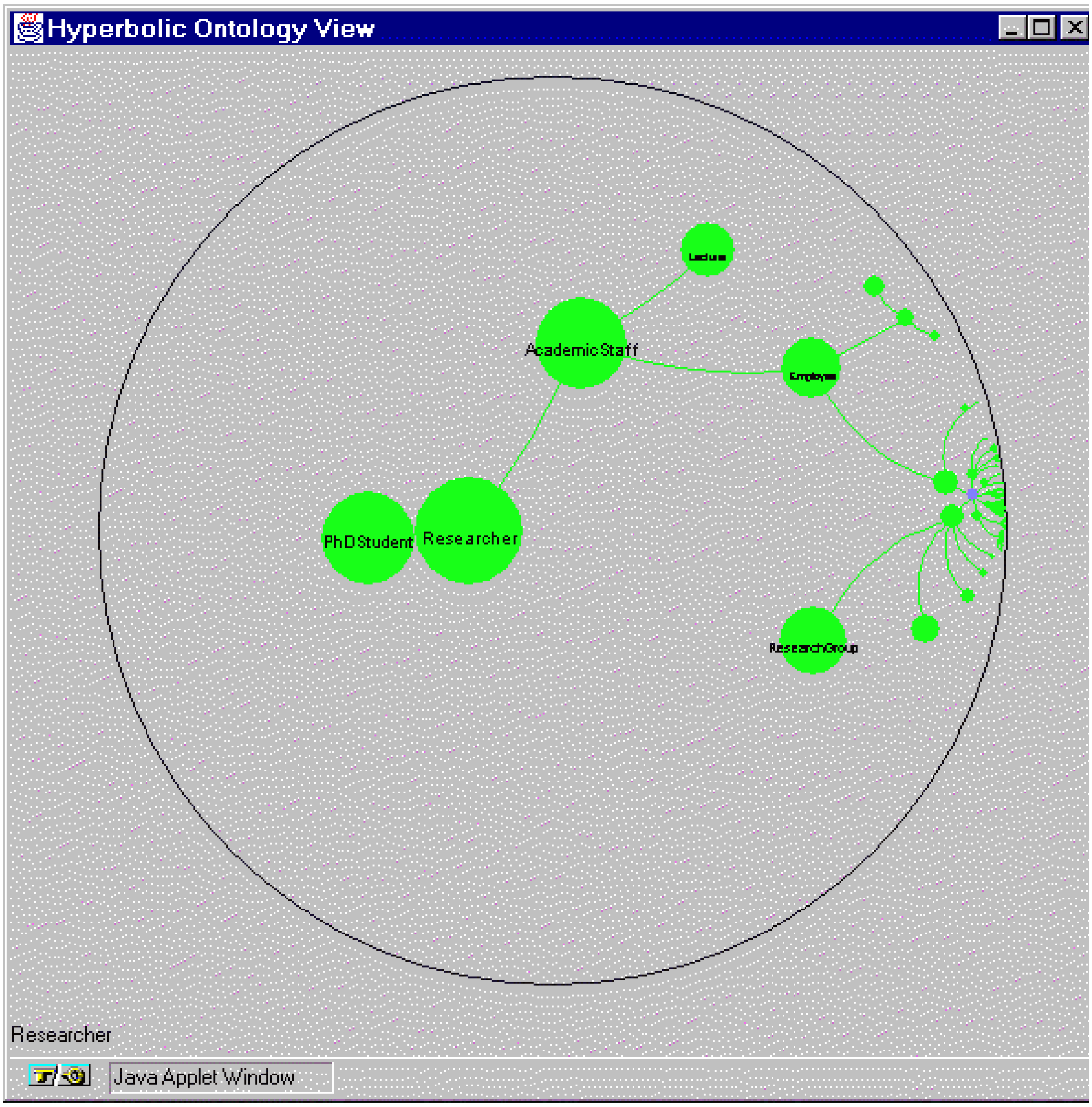


Figure 4 The hyperbolic ontology view.

meantime. The most prominent one is the (KA)^2 initiative[5] that develops an ontology for annotating web documents of the knowledge acquisition community [Benjamins et al., to appear].

---

4. http://www.aifb.uni-karlsruhe.de/www-broker.
5. The Knowledge Annotation Initiative of the Knowledge Acquisition Community (KA)^2.

3 Problems in Scaling Up Ontobroker

Ontobroker produces nice and convincing results for small-sized applications. However, there are serious bottlenecks when trying to apply it to larger case studies. Three main problems significantly reduce its usability. First, there is a high human annotation effort. It is hard to convince customers that this effort really pays back through the improved information access. Second, the inference engine becomes slow when the number of facts increases to realistic sizes. Then queries are no longer get answered or the answers arrive too late. Third, neither the query interface nor the way answers are presented fulfil the needs of information retrieval tools. We will discuss each of these problems in more detail in this section.

3.1 The Annotation Effort

Manually adding annotations to web sources requires human effort causing costs in terms of time and money. In an experiment we estimated that the average number of pages a person can annotate is around five per hour. Obviously this number significantly differs with the size of the pages, the quality, and the grainsize of annotations. Meanwhile we also developed a click-and-browse editor that lowers this effort (see Figure 5).[6] Still, annotating the entire web would require an unrealistic effort. This annotation effort may become less problematic by spreading it over the entire web community. Currently the Resource Description Framework (RDF) [Miller, 1998] arises as a standard for annotating web sources with machine-processable metadata.[7] Relating our approach to this standard significantly broadens the range of existing annotations it can be applied to. Another interesting possibility is the increased use of the eXtensible Markup language XML. In many cases, the tags defined by a Document Type Definition (DTD) may carry semantics that can be used for information retrieval. For example, assume a DTD that defines a person tag and within it a name and phone number tag.

```
<PERSON>
        <NAME>Richard Benjamins</NAME>
        <PHONE>+3120525-6263</PHONE>
</PERSON>
```

Then the information is directly accessible with its semantics and can be processed later by Ontobroker for query answering. Expressed in Frame logic, we get:

```
url[NAME --> „Richard Benjamins“; PHONE -->+3120525-6263] : PERSON
```

Annotation is a declarative way to specify the semantics of information sources. A procedural method is to write a program (called *wrapper*) that extracts factual knowledge from web sources. [Ashish & Knoblock, 1997] distinguish three types of information

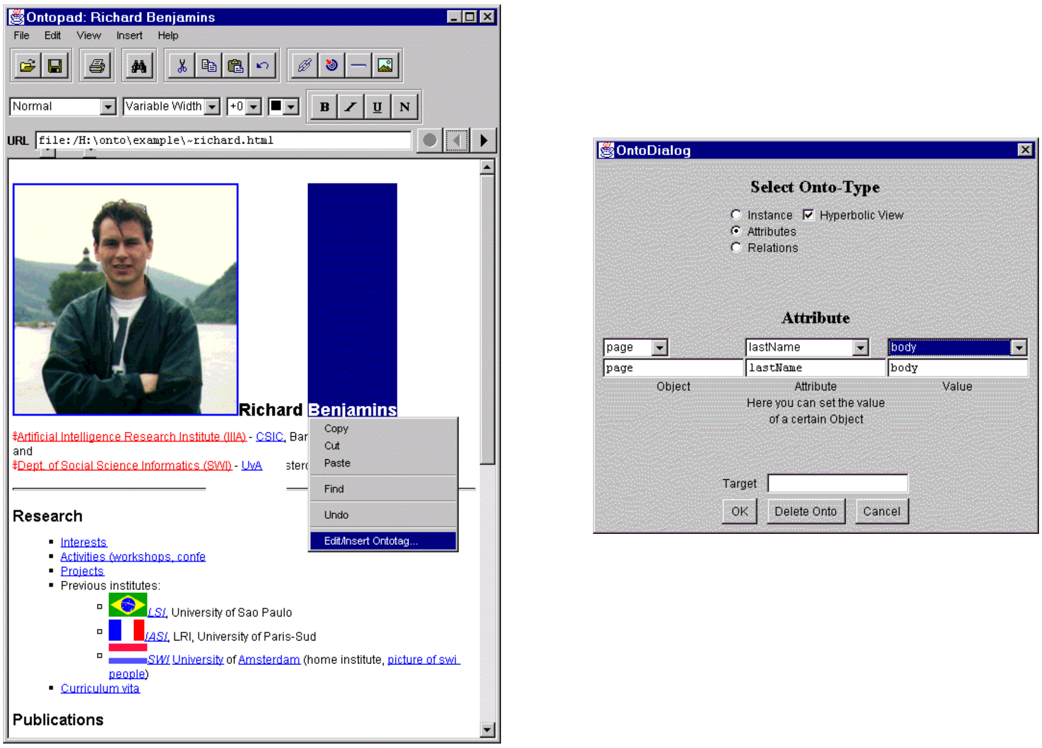
---

[6] The annotation tool presents a HTML page as it is presented by using a Web-browser. It also provides browsing facility for the ontology, where a user can select appropriate terms and the possibility to add annotations to the HTML code using the selected terms from the ontology.

[7] For example, Netscape provides a Yahoo like index and categorized web sources in RDF, see directory.netscape.com.

sources on the web: multiple-instance sources, single-instance sources, and loosely-structured sources. The former two types have a stable format that can be used by a wrapper to extract information. Writing wrappers for stable information sources enable us to apply Ontobroker to structured information sources that do not make use of our annotation language. In fact, we applied Ontobroker to the CIA World Fact book (cf. Figure 6)[8]. The Fact book contains a page for each country in the world which presents some general information using a standardized layout. The wrapper program we developed (a one page phyton program) extracts around 40.000 facts providing around 4 MB of factual knowledge about these countries. The strategy of the wrapper is to use a key word based search combined with assumptions on the delimiters of information entries. This strategy was necessary because the pages are hand made and slightly differ in structure for different countries. Also the authors of these pages used HTML as a layout and not as a logical language.[9]

This experiment proved that it is already possible to exploit structure and regularity in current web sources (i.e., HTML documents) to extract semantic knowledge from it without any additional annotation effort. Successfully overcoming the annotation effort for this information source made us aware of a second bottleneck in Ontobroker, the query response time of inference engine for large sets of facts. We will discuss this



URL file:/H:\onto\example\~richard.html

Richard Benjamins

Artificial Intelligence Research Institute (IIIA) - CSIC, Bar
and
Dept. of Social Science Informatics (SWI) - UvA ster

Research

- Interests
- Activities (workshops, confe
- Projects
- Previous institutes:
  - LSI, University of Sao Paulo
  - IASI, LRI, University of Paris-Sud
  - SWI University of Amsterdam (home institute, picture of swi people)
- Curriculum vita

Publications

Select Onto-Type

○ Instance ☑ Hyperbolic View
● Attributes
○ Relations

Attribute

Object	Attribute	Value
page	lastName	body
page	lastName	body

Here you can set the value
of a certain Object

Target

Figure 5 The annotation editor (it also includes the hperbolic representation of Figure 4).

---

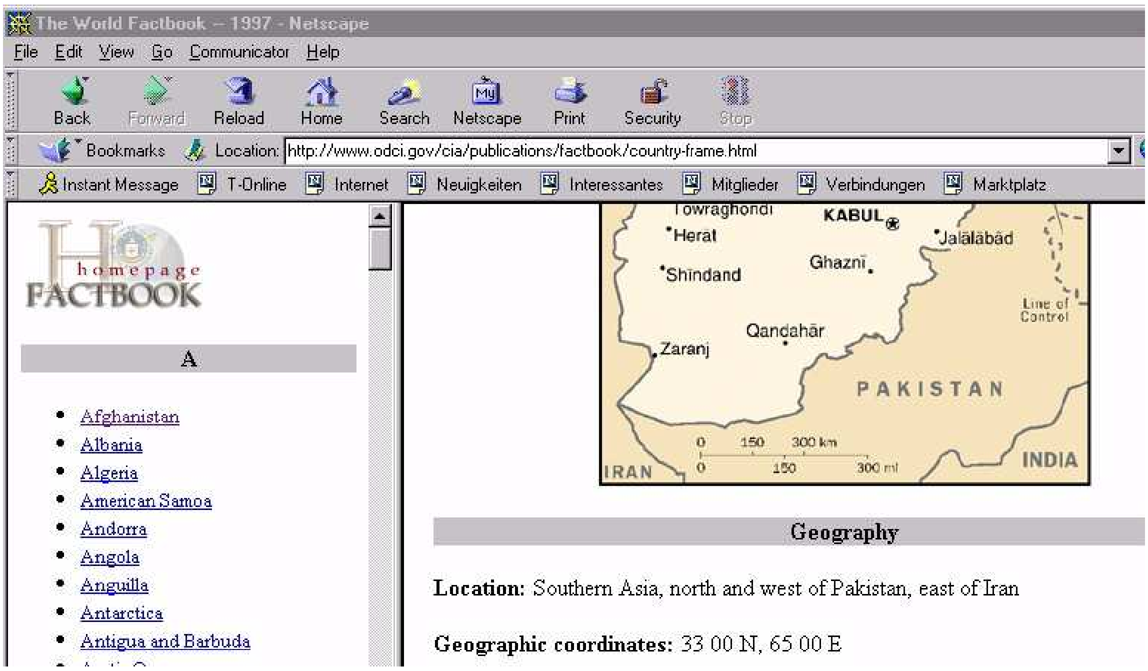
[8] http://www.aifb.uni-karlsruhe.de/www-broker.

[9] For example, they indicate headings by <bold> and not by <Heading$_i$>-tags.

problem in the next section.

3.2 The Inference Effort

In the worst case a query may lead to the evaluation of the entire minimal model. This is a computational hard problem (cf. [Brewka & Dix, 1999]). In other cases predicate symbols and constants are used to divide the set of facts into subsets in order to omit those subsets which can not contribute to the answer. This normally reduces the evaluation effort considerably. Ontobroker allows very flexible queries such as „which attributes has a class“. As a consequence the entire knowledge is represented by only a few predicates such as the predicate *value* which relates a class *c* to its attribute *att* and the corresponding attribute value *v* (*value*(*c*,*att*,*v*)). This reification strategy implies that the set of facts is divided into a few subsets only. The last version of our inference engine used the minimal model as semantics. If the set of rules is stratified [Ullman, 1988] an answer to a query may be evaluated efficiently using minimal model semantics. Using only few predicates has the consequence that nearly every rule set is not stratified if we allow negation in rules. This was the reason that we had to use the Wellfounded Semantics in our approach now (cf. [Van Gelder et al., 1991]). Wellfounded Semantics



http://www.odci.gov/cia/publications/factbook/country-frame.html

Location: Southern Asia, north and west of Pakistan, east of Iran

Geographic coordinates: 33 00 N, 65 00 E