On2broker: Semantic-based access to information sources at the WWW

Dieter Fensel, Jürgen Angele, Stefan Decker, Michael Erdmann, Hans-Peter Schnurr, Steffen Staab, 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/www-broker

Abstract. On2broker provides brokering services to improve access to heterogeneous, distributed and semistructured information sources as they are presented in the World Wide Web. It relies on the use of *ontologies* to make explicit the semantics of web pages. In the paper we will discuss the general architecture and main components of On2broker and provide some application scenarios. We show how we can provide semantic based access to web sources described with HTML, XML, or RDF.

1 Introduction

Meanwhile, the World Wide Web (WWW) contains several hundred million static objects providing a broad variety of information sources. The early question of whether a certain piece of information is on the web has changed into the problem of how to find and extract it. Therefore, dealing with the problem of finding and maintaining information in the WWW has become a highly important topic and several initiatives exist that try to improve the current situation. Currently, most web sources relay on the use of *HTML* that describes structure (and layout) of documents, but not the semantics of the provided information. More recently, *XML* has been proposed that allows the definition of new tags (based on a DTD) that enable the semantic annotation of information (i.e., a tag *person* with subtags *name* and *telephone number* allows direct access to the provided information). Another web standard that is currently under development is the *Resource Description Framework RDF* [RDF] which allows the meta level annotation of web sources making their content explicit and machine processable. In addition to formalisms, standard vocabularies for describing information sources are developed by the *Dublin Core* and *MPEG-7* initiative.

In the paper we describe a tool environment called On2broker that processes information sources and content descriptions in HTML, XML, and RDF and that provides information retrieval, query answering and maintenance support. Central for our approach is the use of *ontologies* to describe background knowledge and to make explicit the semantics of web documents. Ontologies have been developed in the area of knowledge-based systems for structuring and reusing large bodies of knowledge (cf. CYC [Lenat, 1995], $(KA)^2$ [Benjamins et al., to appear]). Ontologies are consensual and formal specifications of vocabularies used to describe a specific domain. Ontologies can be used to describe the semantic structure of complex objects and are therefore well-suited for describing heterogeneous, distributed and semistructured information sources.

On2broker provides a broker architecture with four elements: a query interface for formulating queries, an info agent used for collecting the required knowledge from the web, an inference engine used to derive answers, and a database manager used to cache semantic annotations. On2broker uses semantic information for guiding the query answering process. It provides the answers with a well-defined syntax and semantics that can be directly understood and further processed by automatic agents or other software tools. It enables a homogeneous access to information that is physically distributed and heterogeneously represented in the WWW and it provides information that is not directly represented as facts in the WWW but which can be derived from other facts and some background knowledge. Still, the range of problems it can be applied to is much broader than information access and identification in semistructured information sources:

- *Automatic document generation* extracts information from weakly structured text sources and creates new textual sources. Assume distributed publication lists of members of a research group. The publication list for the whole group can automatically be generated by a query to On2broker. A background agent periodically consults On2broker and updates this page. The gist of this application is that it generates semistructured information presentations *from* other semistructured ones.

- *Maintenance* of weakly structured text sources helps to detect inconsistencies among documents and to detect inconsistencies between documents and external sources, i.e., to detect incorrectness (for example, a publication on a page of a member of the group must also be included in the publication list of the entire group). Again such a service can be provided by On2broker using its ontological representation language and inference engine.

The content of the paper is organized as follows. Section 2 discuss its core architecture and section 3 to 6 its elements. Finally, conclusions, related and future work are given in section 7.

2 The General Picture

The overall architecture of On2broker is provided in Figure 1 which includes four basic engines representing different aspects.

- The **query engine** receives queries and answers them by checking the content of the databases that were filled by the info and inference agents.
- The **info agent** is responsible for collecting factual knowledge from the web using various style of meta annotations, direct annotations like XML and in future also text mining techniques.
- The **inference engine** uses facts and ontologies to derive additional factual knowledge that is only provided implicitly. It frees knowledge providers from the burden of specifying each fact explicitly.
- The **database manager** is the backbone of the entire system. It receives facts from the Info agent, exchanges facts as input and output with the inference agent, and provides facts to the query engine.

Ontologies are the overall structuring principle. The info agent uses them to extract facts, the inference agent to infer facts, the database manager to structure the database and the query engine to provide help in formulating queries. 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 between elements of an ontology and its instances. The ontology introduces the terminology that is used to define the factual knowledge provided by information sources on the web. A little example is provided in Figure 2. It defines the class *Object* and its subclasses *Person* and *Publication*. Some attributes are defined and some rules expressing relationships between them, for example, if a publication has a

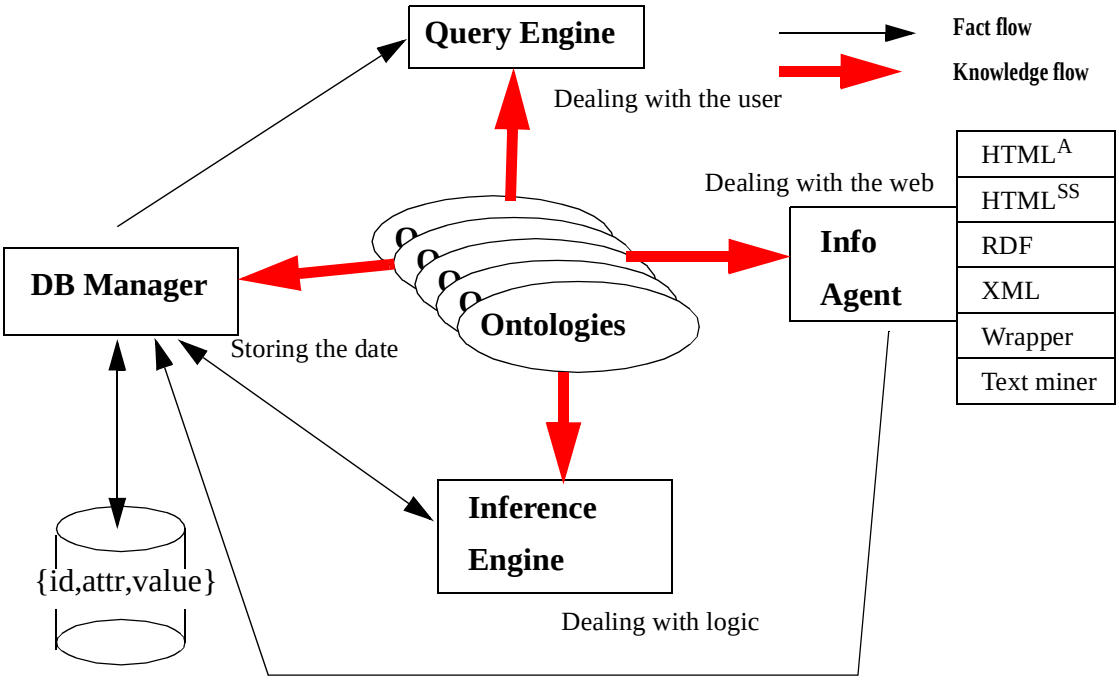

Figure 1 On2brokers Architecture.

```
Object[].
  Person :: Object.
  Publication::Object.
Person[
   Name =>> 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].
```

Figure 2 An excerpt of an ontology (taken from [Benjamins et al., to appear])

person as an author, then the author should have it as a publication.

3 The Query Engine

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* ->> *v*] : *c*

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. 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 and a quick and easy navigation is provided by a presentation scheme based on Hyperbolic Geometry [Lamping et al., 1995] (see Figure 3, and for more details [Fensel et al., 1998a]). Based on these interfaces, On2broker automatically derives the query in textual form and presents the result of the query.

In the effort to create efficient search mechanisms in the WWW, information *mediators*, the like of Metacrawler, and softbots (cf. [Etzioni, 1997]) that access other search engines will become increasingly important. That is why in On2broker the decision was taken to implement the query interface as an Java™ *Remote Method Invocation (RMI) Server.* This allows us to make the Java™ interface publicly available and thus give meta search engines more efficient access to the knowledge base.

4 The Info Agent

The info agent extracts factual knowledge from web sources. We will discuss the four possibilities we provide in On2broker.

First, we developed a small extension of HTML called HTML^A to integrate semantic annotations in HTML documents. On2broker uses a *webcrawler* to collect pages from the web, extracts their annotations, and parses them into the internal format of On2broker. More details on HTML^A can be found in [Decker et al., 1999].

Second, we use wrappers for automatically extracting knowledge from web sources. 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. Writing wrappers for stable information sources enable us to

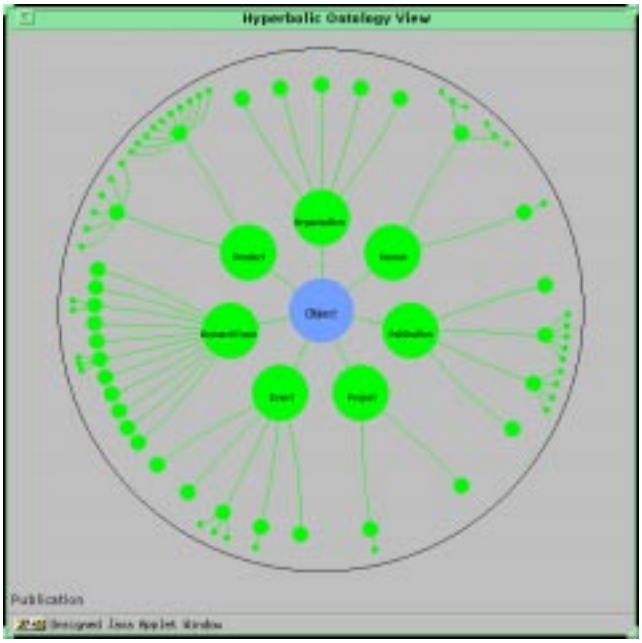

Figure 3 The hyperbolic ontology view.

apply On2broker to structured information sources that do not make use of our annotation language. In fact, we applied On2broker to the CIA World Fact book [CIA]. This shows 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.

Third, On2broker can make use of RDF Annotations (cf. [Lassila & Swick, 1998]). Manually adding annotations to web sources requires human effort causing costs in terms of time and money. However, this annotation effort may become less problematic by spreading it over the entire web community. Currently the Resource Description Framework (RDF) [Lassila & Swick, 1998] arises as a standard for annotating web sources with machine-processable metadata. *RDF* provides means for adding semantics to a document without making any assumptions about the internal structure of this document. The info engine of Onto2broker can deal with RDF descriptions. We make use of the RDF Parser SiRPAC that translates RDF descriptions into triples that can directly be put into our database. More details on how our inference engine works with RDF are given in [Decker et al., 1998]. The inference engine of On2broker specialized for RDF is called *SiLRI (Simple Logic-based RDF Interpreter)*. Actually, On2broker is the first inference engine for RDF.[1]

Fourth, another interesting possibility is the increased use of the eXtensible Markup language XML. In many cases, the tags defined by a 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

RDF still requires the annotation effort for creating metadata but this effort is now shared by the entire web community. XML provides the chance to get metadata „for free“, i.e. as side product of defining the document structure. XML allows the definition of new tags with the help of a DTD and provides semantic information as a by-product of defining the structure of the document. A DTD defines a tree structure to describe documents and the different leaves of the tree have tags that provides semantics of the elementary information units presented by them. That is, the structure and semantics of a documents are interleaved. On2broker is able to read such DTD, to translate it into an ontology, and to translate XML documents into its internal triple representation. Actually, DTDs are serialized and simple means for describing ontologies. The above given example corresponds to a class

1. http://www.w3.org/RDF, RDF Software and Products.

Person with two attributes *Name* and *Phone*. Weaknesses of DTDs from an ontological point of view are:

- No inheritance is provided. Therefore attribute definitions need to be repeated. If classes like *Person*, *Employee*, and *Student* share the same attributes their definition must be repeated for each class making maintenance difficult. In our approach, they can be defined for *Person* and inherited by *Employee* and *Student*.
- DTDs do not provide means to define the range of attributes, i.e. to constrain the values a tag may have.
- DTDs do not provide rules that allow the implicit representation of data. All data need to be represented explicitly (i.e., materialized).

5 The Inference Engine

The *inference engine* takes the facts collected by the webcrawler 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 Horn logic via Lloyd-Topor transformations [Lloyd & Topor, 1984]. Techniques from deductive databases are applicable to implement the second stage: the bottom-up fixpoint evaluation procedure. We have adopted the well-founded model semantics [Van Gelder et al., 1991] and compute this semantics with an extension of dynamic filtering [Van Gelder, 1993].

The inference engine is used to derive information that is implicitly present in web sources without requiring that all information is complete materialized by annotations. We will briefly illustrate this with some examples.

(1) Employee :: Person

This is-a relationship implies that each employee is also a person and each attribute that is defined for persons can also be applied to employees.

(2) X [cooperates ->> Y] -> Y[cooperates --> X]

This rule ensures symmetry of cooperation. If it is known from an annotation that *Motta* cooperates with *Chandrasekaran* than we already know that *Chandrasekaran* cooperates with *Motta*, even if *Chandrasekaran* does not explicitly annotate his homepage with this fact.

(3) X [author ->> Y] : Publication -> Y[has-publication ->> X] : Researcher

The final example states that if in a publication file somebody is stated as author, we will get this publication if we query his homepage for his publications.

Without such an inference mechanism, the provided information is notoriously incomplete and annotation effort is unmanageable high.

6 The Database Manager: Decoupling Inference and Query Response

In the design of Ontobroker (cf. [Fensel et al., 1998a]) we already made an important decision when we separated the web crawler and the inference engine. The web crawler periodically collects information from the web and caches it. The inference engine uses this cache when answering queries. The decoupling of inferencing and fact collection is done for efficiency reasons. The same strategy is used by search engines on the web. A query is answered with help of their indexed cache and not by starting to extract pages from the web. On2broker refines this architecture by introducing a second separation: *separating the query and inference engines*. The inference engine works as a demon in the background. It takes facts from a database, infers new facts and returns these results back into the database. The query engine does not directly interact with the inference engine. Instead it takes facts from the database:

- Whenever inference is a time critical activity, it can be performed in the background independently of the time required to answer the query.
- Using database techniques for the query interface and its underlying facts provides robust tools that can handle mass data.
- It is relatively simple to include things like truncation, term similarity and ranking in the query answering mechanism. They can now directly be integrated into the SQL query interface (i.e., in part they are already provided by SQL) and do not require any changes to the much more complex inference engine.

The strict separation of query and inference engines can be weakened for cases where this separation would cause disadvantages. In many cases it may not be necessary to enter the entire minimal model in a database. Many facts are of intermediate or no interest when answering a query. The inference engine of On2broker incorporates this in its dynamic filtering strategy which uses the query to focus the inference process (cf. [Fensel et al., 1998b]).

7 Conclusions

On2broker is the successor system of Ontobroker (cf. [Fensel et al., 1998a], [Decker et al., 1999]). 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. On2broker is available on the web and has been applied in a number of applications in the meantime. The most prominent one is the $(KA)^2$ initiative that provides semantic access to all kinds of information of research groups of the knowledge acquisition community [Benjamins et al., to appear].

The use of *one* ontology for annotating web documents will never scale up for the entire web. Neither will an ontology be suitable for all subjects and domains nor will ever such a large and heterogeneous community as the web community agree on a complex ontology for describing all their issues (like there will be not one DTD for all purposes). Therefore, work on relating and integrating various ontologies will become an interesting and necessary research enterprise which will also be addressed in the future course of the On2broker project.

References

[Benjamins et al., to appear] R. Benjamins, D. Fensel, S. Decker, and A. Gomez Perez: Knowledge Management Through Ontologies. To appear in the *International Journal of Human-Computer Studies (IJHCS)*.

[CIA] http://www.odci.gov/cia/publications/factbook.

[Decker et al., 1998] S. Decker, D. Brickley, J. Saarela, and J. Angele: A Query and Inference Service for RDF. In *Proceedings of the W3C Query Language Workshop (QL-98)*, Boston, MA, December 3-4, 1998.

[Decker et al., 1999] S. Decker, M. Erdmann, D. Fensel, and R. Studer: Ontobroker: Ontology based Access to Distributed and Semi-Structured Information. In R. Meersman et al. (eds.), *Semantic Issues in Multimedia Systems*, Kluwer Academic Publisher, Boston, 1999.

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

[Fensel et al., 1998a] D. Fensel, S. Decker, M. Erdmann, and R. Studer: Ontobroker: The Very High Idea. In *Proceedings of the 11th International Flairs Conference (FLAIRS-98)*, Sanibal Island, Florida, USA, 131-135, Mai 1998.

[Fensel et al., 1998b] D. Fensel, J. Angele, and R. Studer: The Knowledge Acquisition And Representation Language KARL, *IEEE Transactions on Knowledge and Data Engineering*, 10(4):527-550, 1998.

[Van Gelder, 1993] A. Van Gelder: The Alternating Fixpoint of Logic Programs with Negation, *Journal of Computer and System Sciences*, 47(1):185—221, 1993.

[Van Gelder et al., 1991] A. Van Gelder, K. Ross, J. S. Schlipf: The Well-Founded Semantics for General Logic Programs, *Journal of the ACM*, 38(3): 620—650, 1991.

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

[Lamping et al., 1995] 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.

[Lassila & Swick, 1998] O. Lassila and R. R. Swick (eds.): Resource Description Framework (RDF) Model and Syntax Specification, *W3C Working Draft*, August 19, 1998. http://www.w3c.org/TR/WD-rdf-syntax.

[Lenat, 1995] D. B. Lenat: CYC: A Large-Scale Investment in Knowledge Infrastructure, *Communications of the ACM* 38(11), 1995.

[Lloyd & Topor, 1984] J. W. Lloyd and R. W: Topor: Making Prolog more Expressive, *Journal of Logic Programming*, 3:225-—240, 1984.

[RDF] http.//www.w3c.org/Metadata. See http://www.w3c.org/RDF.