# 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, Univ. 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.

## 1. Introduction

In the paper we describe a tool environment called On2broker[1] that processes information sources and content descriptions in HTML, XML, and RDF and that provides intelligent 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., 1998]). 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 (see Figure 1). 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: It can be applied to automatic document generation, to extract information from weakly structured text sources, to create new textual sources, and to maintain weakly structured text sources.

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

## 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.

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 Datalog-like rules to derive additional information.

*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].

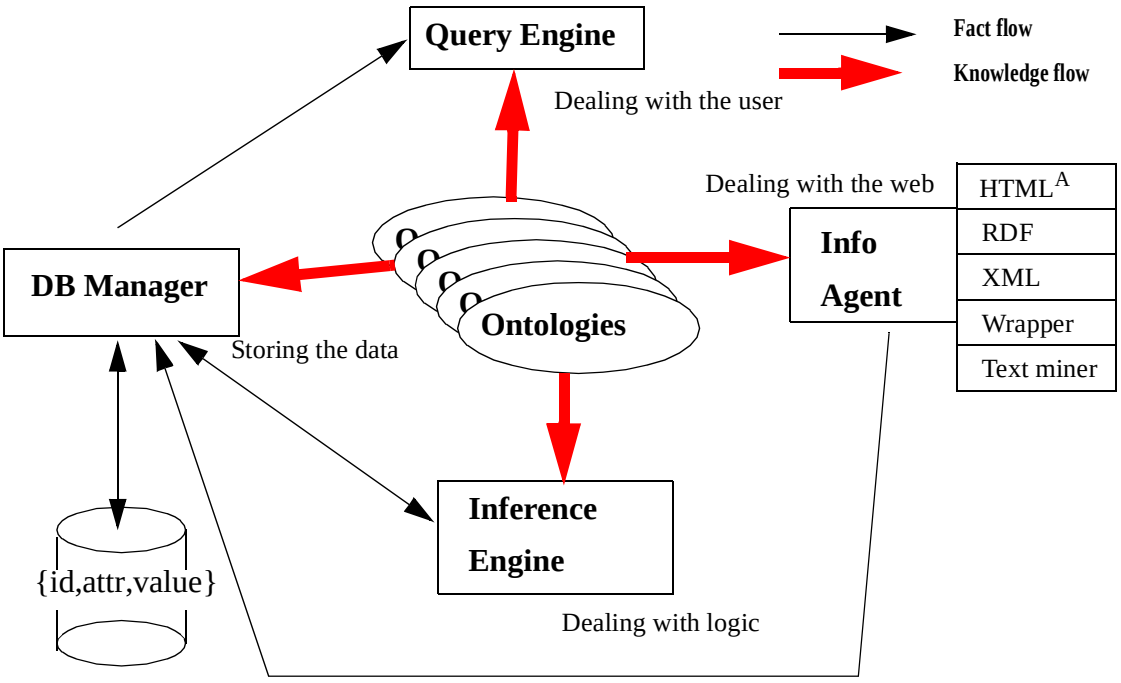


Figure 1 On2brokers Architecture.

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.

## 3. The Query Engine

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

$$x \in c \wedge \text{attribute}(x) = v$$

Complex expressions can be built by combing these elementary expressions with the usual logical connectives. 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 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.

## 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, writing wrappers for stable information sources enable us to 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. 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. *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[1] 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)* and available via http://www.w3.org/RDF/Implementations/SiRPAC/Validation.html.

Fourth, another interesting possibility is the increased

1. http://www.w3.org/RDF/Implementations/SiRPAC/

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. 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. Actually, DTDs are serialized and simple means for describing ontologies.

A limitation of the current version of the info agent is that it does only apply to static web sources, i.e., it cannot access knowledge that is provided by cgi-scripts, JavaScripts or other means that dynamically generate information.

## 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. It is used to derive information that is implicitly present in web sources without requiring that all information is complete materialized by annotations.

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 and compute this semantics with an extension of dynamic filtering [Van Gelder, 1993]. The well founded semantics is the most general, tractable semantics for closed world reasoning with negation [Brewka & Dix, 1998].

We have chosen a closed-world approach, which imply that all answers to queries are only correct with respect to the completeness of the facts known to On2broker. There are other approaches for using negation on the Web [Abiteboul & Vianu, 1997][Himmeröder et al., 1997]. They argue that the web has a open nature and view the Web as infinite. Thus closed world reasoning is not appropriate. They propose to use a version of inflationary semantics: only what is known at the time during the inference process, when negative information is needed, is assumed to be true. Everything else is assumed to be false. The database is dynamically extended on demand during query time by incorporating new web pages to that is already known. Although this open model seems to be quite attractive at the first sight, it has several shortcomings for our requirements:

- Every time a query is posed the engine has to download web-pages from the web. If the connection is slow, this might run a long time.
- Due to the openness of the computation, one has to introduce a bound, when the inference engine has to stop taking information from the web. Otherwise the inference engine might run forever. A typical example for such a border is the border of a web site, e.g. then it is only possible to search for information inside one particular web site.
- Examples for exploiting this open semantics given by [Himmeröder et al., 1997] use links defined in web-pages for loading new pages at query time. This is not usable when using other semantic concepts.
- Declarative queries are not possible with inflationary semantics: what is false during one step of the inference process might be true in another.

Thus inflationary semantics is useful when doing on-line queries in a scope where the exact pages to query are not known at query time and it is necessary to dynamically add pages to the database. We, however, apply On2broker to a subset of the WWW captured by the info engine and apply closed-world assumption to this fragment of the WWW. That is, closed-world assumption is applied in the context generated through the info engine of On2broker.

## 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 the architecture of Ontobroker 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., 1998]. With WebMaster [van Harmelen & van der Meer, 1999], it shares the use of ontologies for improving access and maintenance of web sources. However, the latter is limited to XML sources and its constraint language has less expressive power (horn logic without negation).

SHOE [Luke et al., 1997] introduced the idea of using ontologies for annotating web sources. There are two main differences to our approach. First, the annotation language is not used to annotate existing information in web pages, but to add additional information and annotate them. That is, in SHOE the same information must be repeated and this redundancy may cause significant maintenance problems. For example, an affiliation must once be provided as a text string rendered by the browser and a second time as annotated meta information. A second difference is the use of inference techniques and axioms to infer additional knowledge. SHOE relies only on database techniques. Therefore, no further inference service is provided. On2broker takes an intermediate position. It uses an inference engine to derive additional facts. Its query interface is, however, coupled to a database easily scaling up to large datasets.

**Acknowledgement**. We thank the reviewers of the III99 workshop of IJCAI-99 for their very helpful comments.

## References

[Abiteboul & Vianu, 1997] S. Abiteboul, V. Vianu: Queries and Computation on the Web. In: F. N. Afrati, P. Kolaitis (Eds.): *Database Theory - ICDT '97, 6th International Conference*, Delphi, Greece, January 8-10, 1997, Proceedings. Lecture Notes in Computer Science, Vol. 1186, Springer, 1997, pp. 262-275.

[Benjamins et al., 1998] 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.

[Brewka & Dix, 1998] G. Brewka and J. Dix. Knowledge representation with logic programs. In J. Dix, L. Pereira, and T. Przymusinski, editors, *Logic Programming and Knowledge Representation*, LNAI 1471, pages 1-55, Berlin, 1998. Springer.

[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.

[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)*, Sanibel 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 Harmelen & van der Meer, 1999] F. van Harmelen and J. van der Meer: WebMaster: Knowledge-based Verification of Web-pages. In *Proceedings of the Second International Conference on The Practical Applications of Knowledge Management (PAKeM99)*, London, UK, April 1999, pp. 147-166.

[Himmeröder et al., 1997] R. Himmeröder, G. Lausen, B. Ludäscher, C. Schlepphorst: On a Declarative Semantics for Web Queries. In: F. Bry, R. Ramakrishnan, K. Ramamohanaroa (Eds.): *Deductive and Object-Oriented Databases*, 5th International Conference, DOOD'97, Montreux, Switzerland, December 8-12, 1997, LNCS , Vol. 1341, Springer, 1997

[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.

[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.

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