A Modest Proposal: Reasoning Beyond the Limits of Ontologies

Wolfgang Wohner
Bavarian Research Center for Knowledge Based Systems
Orleansstraße 34, 81667 Munich, Germany
wohner@forwiss.de

Abstract

We will present an approach that extends the formal model of ontologies by application semantics. The novel notion of *laws* governing these semantics is motivated and introduced.

1 Introduction

In this short paper we want to stress the need for a framework that models the inference semantics of ontologies. An ontology provides a formalization of the concepts of an application area and their semantics, indicated e.g. by relations or axioms, but it is lacking a description of how this knowledge may be used for automated reasoning. We suggest to regard ontologies as static formal models that require additional information, i.e. *metadata on ontologies*, in order to be processed correctly. For explanatory purposes we will consider an exemplary ontology used for intelligent searching in semi-structured documents. An extended example in section 2 will motivate considerations about the semantics the ontology has to cover. Section 3 discusses how this knowledge may be applied when processing user queries and argues that an explicit modeling of the underlying patterns and rules is necessary.

2 Seeking Wisdom

Suppose a computer scientist expert is looking for some specific information, say, about the nature of knowledge. As this is a very complex question she might want to consult a local philosopher. The only philosopher living nearby she has heard of is a Mr. Smith but, unfortunately, he is not listed in the phone book. Now, she is looking for his address and so it is only reasonable that she will try to find Web documents containing this information.

Most probably she will first use one or more *keyword-based search engines* such as Google or AltaVista. The computer scientist's task consists of finding adequate keywords to formulate her query. Although her actual interest lies in getting in touch with a (any) philosopher living *nearby* she cannot express this fact using keywords. Generally, drawbacks of the keyword-based approach concern (i) the limited expressiveness of the query languages and (ii) the insufficient treatment of semantic text properties such as linguistic diversity or contextual semantics.

The computer science expert might therefore turn to *information retrieval (IR)* techniques like text mining and information extraction using wrappers. Although *text mining* techniques have been proven to yield acceptable results in certain application areas they are still very limited as they are predominantly concerned with exploiting linguistic features and not with the actual semantics of the text itself. Existing semantic analysis methods are less advanced and computationally too expensive to be used for exhaustive searching in large text corpora [Tan, 1999]. *Wrappers* on the other hand are used for selectively extracting textual components. But wrappers are highly specialized and will return useless results from pages (even valid ones) not complying to their templates, they focus on syntactic structure, not content and, consequently, wrappers know no mechanisms for adapting to different document structures as it is the patterns of these very structures (and not the associated concepts) they are looking for.

In summary, all approaches mentioned so far are lacking:

- a semantic notion of the components of a query (e.g. that 'Smith' is a name)
- a semantic notion of what the query expects as a return value (e.g. an address)
- a technique for adequately processing queries (e.g. adaptively, by semantic query rewriting)
- a general means for extracting the required information from heterogeneous text sources

Common to all of these requirements is the basic need for a sound and explicit modeling of background knowledge. A promising approach can be found in the context of database system design. The information stored in a database is highly structured according to its *schema*, an elaborate abstraction of some application area that has been formalized using e.g. entity/relationship (E/R) techniques. Each data unit of a database is strictly typed, e.g. (using relational syntax) the name 'Smith' might be a string value of an attribute *surname* that appears in a relation called *philosophers*. An according database schema then allows for queries like (supposing the *philosophers* relation also contains an attribute *address*):

```
SELECT   address
FROM     philosophers
WHERE    surname = ‘Smith’
```

Thus the internal structure of a database as depicted by its schema offers powerful querying possibilities: concepts like *surname* can be addressed directly and their semantics are known from the database system design. Nevertheless, there remains a remarkable gap between the homogeneous and well-structured data inside a database system and the heterogeneous, at best semi-structured sources of information found elsewhere, which renders integrating their semantics a complicated and complex task.

Heterogeneity, here, refers to differences in both, internal structure and vocabulary of the documents containing information. Ultimately, the gap between syntax and semantics has to be bridged. This can be facilitated significantly by taking advantage of the properties of markup languages (HTML, XML, SGML) that are used to describe metadata which is structuring and commenting on the textual content of documents. Metadata by itself cannot be directly identified with semantics (after all metadata is still data) but (i) it conforms to a predefined vocabulary and (ii) exhibits structural properties (e.g. nested structures) and these characteristics can be exploited to derive semantics.

The foundations for processing factual knowledge are addressed in the field of ontology engineering. *Ontologies* comprise an abstract knowledge representation of a certain domain. Modeling primitives are concepts, relations, functions, axioms and instances [Gruber, 1993] which are used to formalize the static aspects of the respective domain. There are two general approaches to combine ontologies and markup languages: (i) defining new markup which is directly related to the ontology or (ii) translating foreign markup into native concepts of the local ontology. The first approach has been propagated by SHOE [Luke and Heflin, 2000] and Ontobroker [Fensel *et al.*, 1998] but its drawback is obvious. Since their markup methods did not evolve to become widely accepted standards, only a small portion of Web documents comply with them. For this reason current research, e.g. [Fensel *et al.,* 2000], [Farquhar, 1996], [Stuckenschmidt and Wache, 2000], is focused on establishing a direct linking between domain knowledge and various ways to express it because this provides the basis for reasoning on information which is distributed over a heterogeneous environment such as the semi-structured document space of the Web. The remainder of this paper will motivate a framework that is aimed at providing a formal basis for such reasoning processes which, eventually, could help the computer science expert find the philosopher’s address.

3 Paving the Path

In this section we will examine dynamic aspects of ontology processing. An exemplary system used for providing access to heterogeneous semi-structured data sources will illustrate our approach. Basic assumptions about the system are:

- The system possesses a global ontology that comprises formalized knowledge about a domain.
- There is a set of heterogeneous semi-structured documents (e.g. XML documents) covering topics of that domain.
- There exists a mapping between markup tags of the documents and the concepts of the ontology, i.e. the ontology can ‘understand’ markup semantics in a sense that the concepts involved are part of its formal model.

The system’s main purpose is to answer user queries about the contents of the documents. Return values can be document fractions (e.g. concepts, their values or combinations thereof) or complete documents. In order to retrieve valid results the system first has to understand the semantics of the query and then make use of the ontology’s domain knowledge for exploring the syntactic structures of the documents. The general task is to derive information (semantics) from semi-structured data (data conforming to syntax). There are some properties of semi-structured data the system may take advantage of. We will illustrate this by referring to XML syntax:

- *Syntax definition*: the syntax definition of markup elements used within an XML document is known via its DTD, so the system is aware of all element names, their attributes and subelements.
- *Concepts*: the semantics of the structuring elements (tags) are known to the system because of the mapping between elements and ontology concepts.
- *Context*: markup elements are organized hierarchically thus establishing contexts (e.g. by nesting tags like <Name> and <Address> into <Person>) which can be interpreted semantically.
- *Types*: in a weak sense each markup element represents a type of its own but it is also possible to introduce primitive or derived element datatypes using e.g. XML Schema.

This syntax information can be utilized when processing queries that work on semi-structured documents. Existing systems, like On2broker [Fensel *et al.,* 2000], that provide access to semi-structured information sources are dealing with this task but the inference mechanisms and heuristics applied here are usually hidden within their software components. We want to stress the importance of uncovering the underlying semantics and integrating them into the ontology structure. This is not just a matter of rendering implicit processes explicit but of providing a formal semantic model *about the usage* of the semantics an ontology provides on its part. Thus, such a formalization defines *metadata* about the ontology, foremost semantic processing rules we call *laws*. Again, laws have to be understood and executed by software components but the invaluable benefit they could provide is a homogeneous formal description of the semantic and syntactic implications of such processes.

Laws may be regarded as function templates that accept *cases* (e.g. a query) and contain formalized descriptions how to solve them. Our framework is aimed at defining a theoretical basis for such ontology laws and their impact on other elements of the ontology. For the remainder of this section we will stress various aspects of laws by referring to the illustrative example of the previous section.

- Laws address inference semantics.

The original query, '*Find the address of a philosopher living nearby'*, contains an inexact, or *vague*, concept: *nearby*. The meaning of *nearby* depends on the context of the query, as there are different notions of closeness in the context of houses and, say, atoms. In such cases techniques are needed to establish context which requires laws that describe how the desired information can be deduced. These techniques may vary for different semantic classes, or *categories*, of concepts, such as precise and vague ones, i.e.

- Laws can be general or attributed to single concepts or concept categories.

It is of major importance to identify such categories in order to establish a formal basis for reasoning processes. Once the category of a concept is known all laws attributed to that category can be directly applied to this concept as well.

- Laws state the limits of ontologies.

Some knowledge cannot be deduced because of incomplete knowledge. Although the context of *nearby* may be correctly inferred the point of reference (e.g. the computer scientist's own address) remains unknown. This indicates incomplete knowledge about the defining constituents of the query, i.e. at least one input factor of the respective law is missing and there is no other law describing how to compute it. Similarly, the ontology itself might be lacking concepts as well, e.g. a notion for closeness within the context of addresses might not be included. Generally, laws address *representational limits*, i.e. what can be expressed by an ontology, and *inferential limits* about what can be deduced from these representations.

- Laws control semantic query rewriting.

Automated *semantic query rewriting* is a promising technique for improving query return values. Using ontology knowledge an original query may be transformed into a set of refined queries. The excerpt of an XML document shown below does not contain an <Address> tag, so a query restricted to searching addresses would omit this document:

```
<Person>
   <Name> Smith </Name>
   <Phone> (222) 333-4444 </Phone>
   <Profession> philosopher </Profession>
</Person>
```

By contrast, laws provide rules for extending the scope of the query from addresses to e.g. phone numbers, street names and other address components known to the ontology. This would yield Mr. Smith's phone number, valuable information that the original query could not have produced.

- Laws manage uncertainty.

Uncertainty may play an important role in the context of iterative document querying, i.e. reasoning on grounds of intermediate results extracted from texts. From the XML example shown above it can be inferred that 'philosopher' is an instance of the concept *profession*. The value 'philosopher' can now be interpreted as a concept as well. But as this information has been derived from the textual content of a document it must be regarded as uncertain knowledge. Markup elements, on the other hand, can be mapped to concepts directly and therefore establish reliable knowledge. Uncertain knowledge is an omnipresent factor in intelligent information management and we will intensify our research efforts in that direction.

4 Conclusions and Future Work

We have motivated the importance of a framework for classifying and representing ontology laws and discussed some possible applications. Our future work will consist of elaborating this approach by providing a sound formal foundation of such a framework and incorporating a basic set of laws into the ontology of the intelligent information management system we are currently developing.

5 References

[Farquhar, 1996] A. Farquhar, R. Fikes, J. Rice. The Ontolingua Server: A tool for Collaborative Ontology Construction. *Proceedings of KAW96*. Banff, Canada, 1996.

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

[Fensel *et al.*, 2000] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, S. Staab, R. Studer, A. Witt. On2broker: Semantic Access to Information Sources at the WWW. In *Proceedings of IJCAI-99 Workshop on Intelligent Information Integration,* Stockholm, 31 July 1999.

[Gruber, 1993] R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition #5*, pp. 199-200, 1993.

[Luke and Heflin, 2000] Luke, S., Heflin J. SHOE 1.01. Proposed Specification. SHOE Project. February 2000. http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm

[Stuckenschmidt and Wache, 2000] Context Modeling and Transformation for Semantic Interoperability. In *Proceedings of the 7th International Workshop on Knowledge Representation meets Databases (KRDB 2000)*, Berlin, Germany, August 21, 2000.

[Tan, 1999] A.-H. Tan. Text Mining: The state of the art and the challenges. In *Proceedings of the PAKDD'99 workshop on Knowledge Discovery from Advanced Databases*, Beijing, pp. 65-70, April 1999.