Semantic modeling using RDF

Contents

Introduction

Semantic modeling means that the **concepts** of a certain domain are described in a formal way. In this paper, we shall explain semantic modeling using RDF. The acronym **RDF** comes from the terms Resource Description Framework. RDF is a data model and a World Wide Web Consortium (W3C) specification. The syntax of RDF is often expressed using **XML**. However, because RDF is a data model, also other syntactic representations besides XML are possible.

In fact, when we describe semantic modeling in this paper, it will be done using **RDF Schema (RDFS)**, and not just plain RDF. RDFS is a schema language that the users may use to define their own concepts in a particular domain. The relation between RDF and RDF Schema is not similar to the relation between XML and XML schema. XML Schema constrains the structure of XML documents, whereas RDF Schema defines the vocabulary that is used in RDF data models. In RDFS, we can define concepts, the properties that the concepts may have, the values that the properties may take, and the relationships between them. RDFS itself is expressed using the RDF data model.

Let us now briefly introduce the basic notions of the RDF data model. Its basic building block is a **resource – property-value triple**, called a **statement**. These triplets may be represented as directed graphs. Figure 1 shows an example triplet. Eric is the resource, wrote is the property and book is the value. (Antoniou and Harmelen, 2008, W3CRDF).

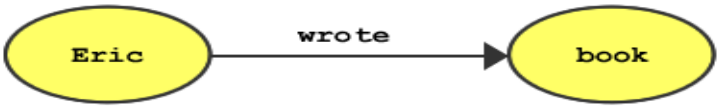

Figure 1: A simple RDF triplet represented as a directed graph.

Resources can be thought of as the subject of a sentence, properties as the predicate and values as the object. Every resource has an **URI**, a Uniform Resource Identifier. A URI may be a URL, Uniform Resource Locator or Web address. Properties are resources that describe relations between resources. They themselves are also identified by URIs. Using URIs to identify resources gives us a global, unique naming scheme. This removes the problem of unique identifiers in distributed data representation. The third item of an RDF triplet is the value. A value may be a resource or a literal. Note that when a RDF model is represented as a directed graph, or as a set of directed graphs, all algorithms for processing this data structure are available.

Let us now give a simple example of an RDF schema using the same graph representation.

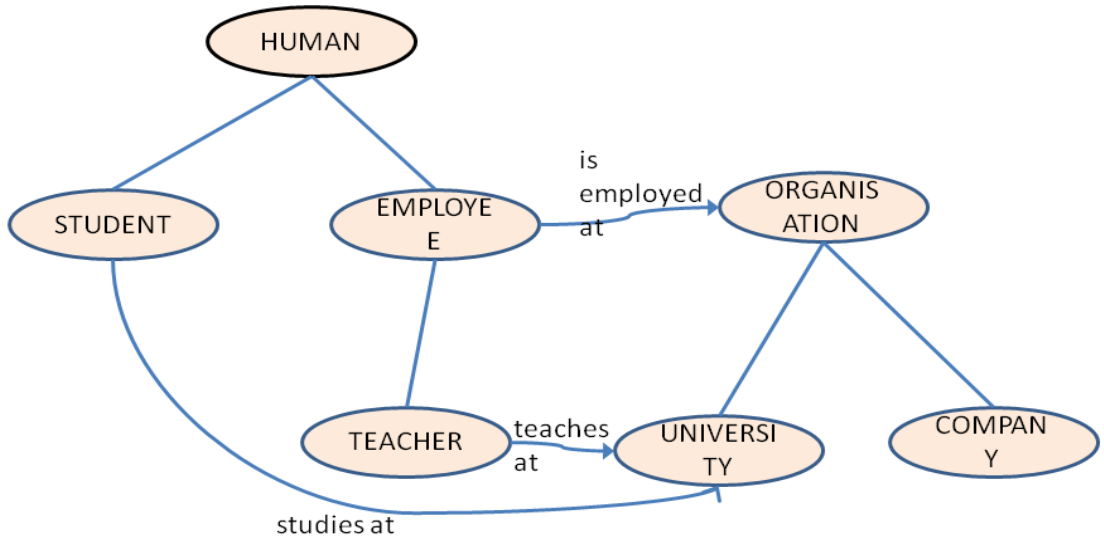

Figure 2: A simple RDF schema expressed as a directed graph. The class hierarchy is expressed by simple lines between nodes, e.g. Teacher **belongs to the class** Employee**. Properties are denoted by arrows from one node to another, e.g.** Teacher **has the property** teaches at**.**

In Figure 2, there is a RDF schema that models the actors in a learning activity in a university setting. The schema models the types of objects in a certain domain, but it does not say anything about the individual objects, or **instances**. The concepts of the schema are illustrated by the ovals in Figure 2, and they include objects such as Human and Student. The concepts are called **classes**. The relationship between the instances and classes is defined using the **rdf:type** –property.

In Figure 2, we may also observe that the classes form a hierarchy. For example, Student is a subclass of Human. The property for defining a subclass is rdfs:subClassOf. Since RDFS is expressed using RDF, we could have defined ourselves a resource called subClassOf and we could have defined it to be a property. However, since the property rdfs:subClassOf has already been defined, it makes sense to use it instead.

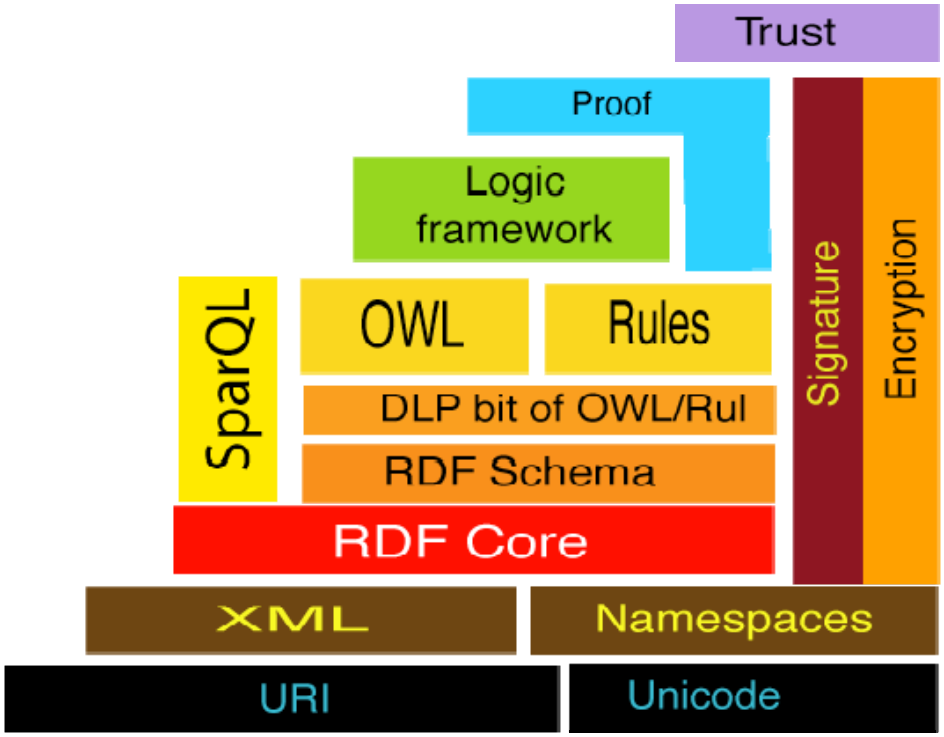

Figure 3: Layers of the semantic web by Tim Berners-Lee

Figure 3 shows the layers of the Semantic Web as seen by the inventor of the World Wide Web. The picture shows the relative position of RDF and RDF Schema. Semantic Web is a vision of such a web whose semantics can be understood by both humans and software agents. Some of the underlying technology has been implemented, but much of it is still under research. The RDF model that is presented in this paper is one of the so called semantic web technologies, even though it can be used in other contexts also (Antoniou and Harmelen, 2008, w3c2004).

Motivation

RDF is a standardized way of representing the semantics of a certain domain. It is a recommendation given by the W3C (w3c2004). Using a standard way of representing semantics enables a better **integration and interoperability** of information systems. In addition, the use of standards is a source for better **reusability**, easier maintenance and quality in the models developed.

In addition to interoperability, the use of the RDF model also enhances the **functionality** of an information system because the data is now enriched with a rich semantic information. This semantic information makes it possible to add new functionalities to the system.

As mentioned earlier in this paper, the domain knowledge of a certain field may be modeled using RDFS. These models may be called ontologies. A common **ontology** provides a way to share a **common understanding** of the structure of information among people and/or software agents. It makes the domain assumptions explicit and it is a means for analyzing domain knowledge. An ontology is a means of separating operational knowledge from domain knowledge. In this paper, we use the term ontology in the following sense. An ontology is:

- a formal, explicit definition of concepts in a domain. The concepts are called classes.
- properties of each concept describing various features and attributes of the concept.

Big ontology creation projects are going on in several fields. In this paper, the term ontology is used in the sense it has in computer science, i.e. it defines a common vocabulary for professionals and/or software agents who need to share

information in a domain. An ontology includes machine-interpretable definitions of basic concepts and relations between them. In other disciplines than computer science, the term ontology typically has a very different meaning. Domain experts would model their domain in any case, and one argument in favor of using RDFS is that it makes sense to use a standardized format instead of a proprietary one (Antoniou and Harmelen, 2008, w3c2004).

Practical Examples

We present here a practical example of using RDF. The publishing company Elsevier uses RDF in order to enable better searches both for its own purposes and for its customers. Elsevier has made its journals available online, but this has not really changed the organization of the product line. Customers at Elsevier can take subscriptions to the online content, but these products are organized along the traditional product lines, i.e. journals that cover distinct fields of science, such as chemistry, biology or medicine. Let us call these traditional products vertical products.

The problem is, that with the rapid developments in several sciences, the traditional division is no longer satisfactory. Customers would like to access journals relating to a certain topic, no matter under which discipline it has been filed. For example, the customer might wish to access all information relating to the Alzheimer's disease, regardless whether this information comes from a biology journal, chemistry journal or medical journal. Thus, the customer is no longer satisfied with the vertical products that are offered, but would rather ask for a horizontal product.

Currently, Elsevier has problems in offering such horizontal products because the information in each journal is organized using different standards. The problems of differences in syntax have been overcome by translating much of the content into an XML format that permits cross-journal querying. However, the semantic problem still remains. Of course, it is possible to search across multiple journals for articles containing the same keywords or perform a free text query, but given the extensive homonym and synonym problems within and between the various disciplines, the results of such a query tend to be unsatisfactory. An example of a homonym problem is the word ontology, that has a different meaning in computer science and in philosophy. An example of a synonym problem is that a user may use a software name, e.g. Google when he is actually interested in search engines in general.

So how does RDF help us in this case? The answer is twofold: First, RDF is used as an interoperability format between heterogeneous data sources, and secondly, the ontology against which the queries are preformed is expressed in RDF. The Figure below illustrates the situation:

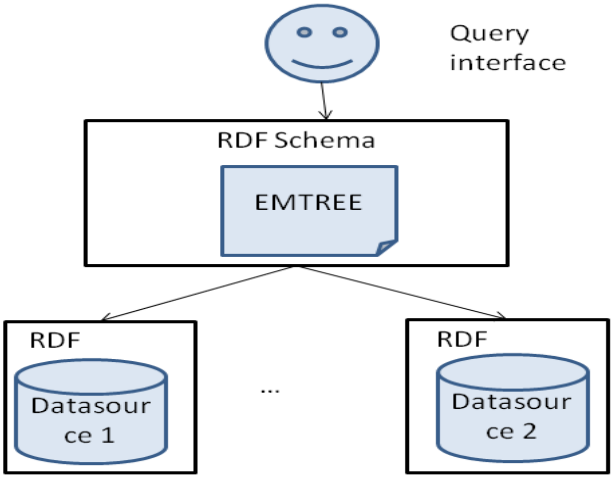

Figure 4: Querying across the data source at Elsevier.

The ontology that is used is the Elsevier's life science thesaurus, EMTREE. It has 42,000 indexing terms and 175,000 synonyms. First, the existing thesaurus has been converted into RDF format. Each of the data sources has then been mapped into this unifying ontology, which is finally used as the single point of entry for all of them. How was this all done in practice? In order to provide the mapping of the journals to the ontology, the concept extraction software from Collexis (http://www.collexis.com) was used. It extracted from the journals the terms belonging to the ontology. The Sesame RDF storage (http://www.openrdf.org) and query engine was used to implement the storage of the ontology data sources as well as the queries. The user interface for searching and browsing was constructed using Aduna ClusterMap software (http://www.aduna-software.com/technology/clustermap).

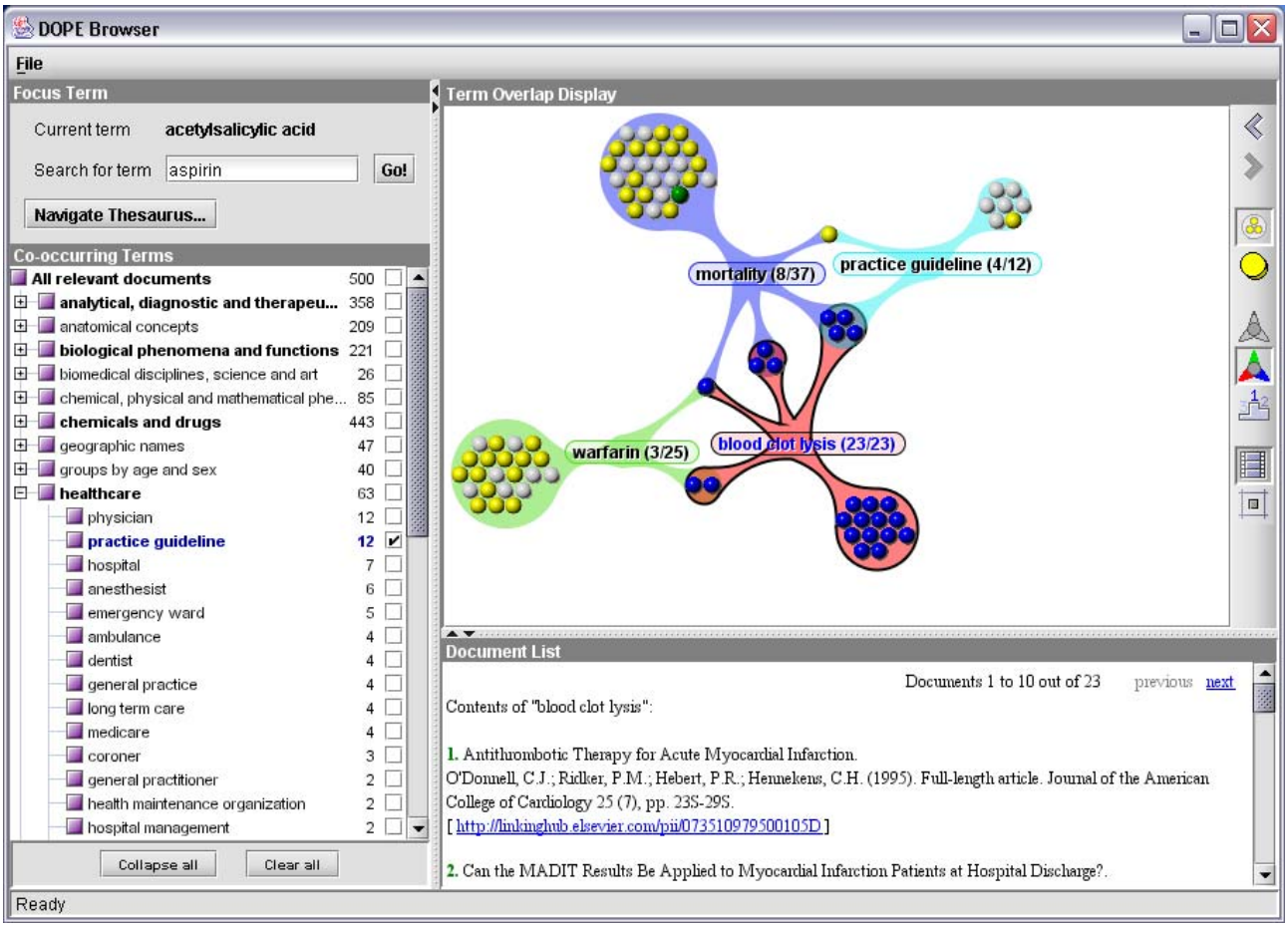

Figure 5: The search and browse interface of the Elsevier digital collection of journals.

Figure 5 shows the user interface of the online journal collection. Let us explain the example it illustrates. Suppose a user wants to browse through the existing literature on aspirin. The string “aspirin” can be entered in the text field at the upper left of the figure. The system then consults Sesame for all keywords that are related to this string. It responds with a dialog showing four possible **EMTREE terms**, asking the user to select one. (This dialog is omitted when there is only one exact match with an EMTREE keyword.) Assuming that the user chooses the keyword “acetylsalicylic acid”, which is the chemical name corresponding with the brand name, this becomes the new **focus keyword**. Now the user may choose **secondary keywords**.

Figure 5 shows the state of the interface after the user has checked the secondary keywords “mortality”, “practice guideline”, “blood clot lysis” and “warfarin”. The visualization graph shows if and how their document sets overlap. Each sphere in the graph represents an individual document, with its color reflecting the document type, e.g. full article, review article or abstract. The colored edges between keyword and clusters of spheres reveal that those documents are indexed with that keyword.

As we can see from the above search and browse user interface example, using an RDF schema as an ontology enables new functionality in the Elsevier digital library system. This functionality includes:

- Disambiguation of the initial keyword queries by the user by detecting the different possible meanings in the ontology
- Organisation of search results hierarchically using the ontology
- Graphical presentation of search results in clusters based on their location in the ontology.

- Widening or narrowing of the queries by navigating up or down the ontology hierarchy. (Waard et al., 2007, Antoniou and Harmelen, 2008)

Ontology authoring using RDF Schema

How is ontology authoring done in practice? The basis of the methods that are used in ontology authoring are the same that are used in software engineering, object-oriented design and knowledge engineering. The process of ontology design may consist of the following steps:

1) Determine scope
2) Consider reuse. Search for existing ontologies e.g. at the Schema Web ontology repository, http://www.schemaweb.info/default.aspx (Schemaweb2009).
3) Find out the concepts. Typically substantives (future resources) and verbs (future properties).
4) Define hierarchy
5) Define properties
6) Test the model by populating it with instances.
7) Check for anomalies

Ontology creation is typically a long process where the experts of a domain gather together in order to create a common understanding of a field. This is why collaborative environments such as collaborative Protégé (Protege2009) for ontology creation have been devised. Also the (semi-) automatic acquisition of ontologies and of the (semi-)automatic population of ontologies has been a very popular field of research. The aim of this research is to at least partially automate the task. (Antoniou and Harmelen, 2008)

Persisting RDF models

There are several freely available frameworks that support RDF storage and querying. Jena and Sesame might be the most known of these.

The Jena Java framework can be used to create and populate RDF models, to persist them to a database, and to query them programmatically using the RDQL query language. The RDBMS supported by Jena are HSQLDB, MySQL, PostgreSQL, Derby, Oracle and Microsoft SQL Server.

Sesame is an open source RDF framework with support for RDF Schema querying and storage of RDF data. Originally, it was developed as a research prototype for an EU research project. Sesame has been designed with flexibility in mind. It can be deployed on top of a variety of storage systems (relational databases, in-memory, filesystems, etc.), and offers a large collection of tools to developers to leverage the power of RDF and RDF Schema, such as a flexible access API, which supports several query languages.

Traditional relational database management systems offer only little or no support for RDF. Out of the big mainstream DBMS providers, only **Oracle** seems to offer some support for native RDF. Oracle Spatial 11*g* is claimed to be an open, scalable, secure and reliable RDF management platform. It uses a graph data model and RDF triples can be persisted, indexed and queried. In Oracle 11g, RDF data is queried using SQL. This is done using a function called SEM_MATCH which is embedded into a SQL query. With the SEM_MATCH function a user may search for an arbitrary pattern against the RDF models, data inferred using RDFS or user-defined rules. According to Oracle, The SEM_MATCH function meets most of the requirements identified by the W3C SPARQL standard for graph queries. As a side note, it is interesting to that in Oracle 11 g also relational data may be queried using semantic operators. These operators are called SEM_RELATED and SEM_DISTANCE. They enable an efficient way of ontology-assisted querying to relational data. The problem with the small and open source RDBMS implementations has been efficiency. With real life size RDFS and real life amounts of data, the systems that use native RDF have been too slow. (Jena, 2009, Sesame, 2009)

References

(Waard et al., 2007) Anita de Waard, Elsevier, Christiaan Fluit, Aduna, and Frank van Harmelen: Semantic Web Use Cases and Case Studies, Use Case: Drug Ontology Project for Elsevier (DOPE), 2007 http://www.w3.org/2001/sw/sweo/public/UseCases/Elsevier/, (as of 1.9.2009).

(w3c, 2004) World Wide Web Consortium's RDF specification http://www.w3.org/RDF/, (as of 1.9.2009).

(Antoniou and Harmelen, 2008) Antoniou, Harmelen: A Semantic Web Primer, 2nd edition, 2008.

(Jena, 2009) The JENA framework. http://jena.sourceforge.net/DB/index.html, (as of 1.9.2009).

(Sesame, 2009) The Sesame framework. http://www.openrdf.org , (as of 1.9.2009).

(Schemaweb, 2009) Repository of RDF Schemas. http://www.schemaweb.info/default.aspx , (as of 1.9.2009)

(Protégé, 2009) Protégé ontology editor. http://protege.stanford.edu , (as of 1.9.2009).

(Oracle, 2009) Oracle Semantic Technologies, Oracle Data Sheet, http://www.oracle.com/technology/tech/semantic_technologies/ , (as of 11.11.2009).

This work has been partly funded by the European Union Lifelong Learning Programme (EU LLP) Transversal Programme.