INTRODUCTION

The World Wide Web today is a huge network of information resources which was built in order to broadcast information for human users. Consequently, most of the information on the Web is designed to be suitable for human consumption: the structuring principles are weak, many different kinds of information co-exist, and most of the information is represented as free text.

With the increasing size of the Web and the availability of new technologies, such as mobile applications or smart devices, there is a strong need for making the information on the World Wide Web accessible to computer programs which search, filter, convert, interpret, and summarize the information for the benefit of the user. The Semantic Web is a synonym for a World Wide Web whose accessibility is similar to a deductive database where programs can perform well-defined operations on well-defined data or even derive new information from existing data (Berners-Lee, Hendler & Lassila, 2001).

One of the main developments connected with the Semantic Web is the resource description framework RDF. RDF is an XML-based language for creating metadata about information resources on the Web. The metadata model is based on a resource which could be any piece of information with a unique name called URI (uniform resource identifier). URIs can either be unique resource locators (URLs)—well known from conventional Web pages—but also tagged information contained on a page or on other RDF definitions. The structure of RDF is very simple: a set of statement forms a labeled, directed graph where resources are represented by nodes and relations between resources by arcs. These are labeled with the name of the relation (Klyne & Carroll, 2004).

RDF, as such, only provides the user with a language for metadata. It does not make any commitment to a conceptual structure or a set of relations to be used. The RDF schema model defines a simple frame system structure by introducing standard relations like inheritance and instantiation, standard resources for classes, relations as well as a small set of restrictions on objects in a relation. Using these primitives, it is possible to define terminological knowledge about resources and relations mentioned in an RDF model (Brickley & Guha, 2004).

BACKGROUND

A characteristic property of an RDF model is that its statements form a labeled, directed graph. The resources mentioned in the statements can be seen as nodes in such a graph (this may include properties as special kinds of resources). Subject and object of each statement are connected by a directed link labeled with the name of the property acting as predicate.

Based on the graph-based view on an RDF model, we can characterize some properties of RDF models that are relevant for query answering. The first basic property of RDF is the fact that a graph entails all its subgraphs. The RDF data model described above and its associated semantics provides us with a basis for defining queries on RDF models in a straightforward way. The idea that has been proposed elsewhere and is adopted here is to use graphs with unlabeled nodes as queries. The unlabeled elements in a query graph can be seen as variables of the query. Answers to a query can be defined as:

- A subgraph of the given RDF model that is an instantiation of the query graph.
- The set of resources used to instantiate unlabeled nodes in the query graph.

From a theoretical point of view, these two definitions are exchangeable as one can easily be derived from the other by either extracting instantiated resources from the answer graph or by instantiating the query graph with the resources from the answer set, respectively.

One of the main features of RDF is the idea to enrich metadata descriptions with explicit models of their intended meaning. These models are defined in terms of a schema definition. We want to exploit this semantic information for query answering as it provides us with background information for computing more complete results. For this purpose, the RDF schema language contains properties with a standardized interpretation.
for defining hierarchies of classes and properties as well as domain and range restrictions for user-defined properties. The intended meaning of these constructs and their impact on an RDF model is defined by the RDF semantics specification. The specification describes a model theoretic semantics and the notion of entailment for arbitrary RDF models (Hayes, 2004). It also defines a set of inference rules that can be used to derive implicit statements from a given RDF model. This implicit information should be taken into account when querying an RDF model making the query processing in the sense of a deductive database. A common approach is to compute the deductive closure of the model to be queried and execute the query on the resulting expanded model.

Figure 1 shows an RDF graph that represents a query. The nodes labeled X and Y are unlabeled nodes in the sense of the RDF model that acts as query variables. Assuming that inProceedings is a subclass of Publication and keyword is a subproperty of about, we see that the graph in Figure 1 represents a result of the query in Figure 2.

RDF QUERY LANGUAGES AND SYSTEMS

A number of RDF query languages have been proposed that implement the general approach mentioned in the last section. Due to space limitations, we cannot discuss all of these languages. We focus on languages implemented in freely available RDF storage and retrieval systems.

JENA and RDQL

The JENA system is a JAVA-based RDF infrastructure for parsing, storing, and accessing RDF data (http://jena.sourceforge.com). JENA is developed by the HP research labs in Bristol and implements the RDQL query language for accessing stored RDF data (http://www.hpl.hp.com/semweb/rdql.htm). An RDQL query consists of a SELECT part that specifies return variables, a WHERE part that refers to the RDF model to be accessed, a FROM part that provides a set of statement patterns that have to be matched by the RDF data in the model, a set of constraints on the variables occurring in the statement patterns, as well as a USING part where abbreviations for XML namespaces can be defined to simplify the query (Seabourne, 2004). In RDQL, schema awareness is not directly integrated in the language specification but has to be provided by the underlying data source.

RDFSuite and RQL

The RDFSuite (Alexaki et al., 2002) is a toolkit for parsing, storing, and querying RDF data developed by ICS-FORTH in Greece (http://139.91.183.30:9090/RDF/). The RQL query language implemented in this suite is probably the most widely known and used RDF query language (http://139.91.183.30:9090/RDF/RQL/). RQL is based on functional query languages for object-oriented databases (in particular, OQL). Different from RDQL, the language provides a rich set of operators for specifying the query result that can freely be combined as a result of the functional nature of the language. These include explicit operators for navigating the schema (retrieving all super-classes or just the direct super-classes). Another distinguishing feature of RQL is the use of path expressions for navigating the RDF graph (Karvounarakis, Christophides & Plexousakis, 2001). This significantly reduces the number of equality constraints that are necessary in languages like RDQL for defining complex statement schemes.

Sesame and SeRQL

The Sesame system (Broekstra, Kampman & van Harmelen, 2002) developed by the Dutch company Aduna is an RDF infrastructure that aims at providing RDF support on top of different information sources such as databases, search engines, and Web services (http://www.openrdf.org/). Besides providing support for the two above-mentioned languages, Sesame also implements its own query language SeRQL (http://www.openrdf.org/publications/users/ch05.html). The SeRQL language
Related Content

Ontology-Supported Web Service Composition: An Approach to Service-Oriented Knowledge Management in Corporate Services
[www.igi-global.com/article/ontology-supported-web-service-composition/3348?camid=4v1a](www.igi-global.com/article/ontology-supported-web-service-composition/3348?camid=4v1a)

Qualifying Objects in Classical Relational Database Querying
[www.igi-global.com/chapter/qualifying-objects-classical-relational-database/20355?camid=4v1a](www.igi-global.com/chapter/qualifying-objects-classical-relational-database/20355?camid=4v1a)

Inclusion Dependencies
[www.igi-global.com/chapter/inclusion-dependencies/8279?camid=4v1a](www.igi-global.com/chapter/inclusion-dependencies/8279?camid=4v1a)

Optimization of Multidimensional Aggregates in Data Warehouses
[www.igi-global.com/chapter/optimization-multidimensional-aggregates-data-warehouses/8040?camid=4v1a](www.igi-global.com/chapter/optimization-multidimensional-aggregates-data-warehouses/8040?camid=4v1a)