INTRODUCTION

Information on the Web has grown very quickly. The semantics of this information are becoming explicit and the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001) is emerging. Ontologies provide a formal representation of the real world by defining concepts and relationships between them. In order to provide semantics to Web resources, instances of such concepts and relationships are used to annotate them. These annotations on the resources, which are based on ontologies, are the foundation of the Semantic Web. Because of the Web’s size we have to deal with large amounts of knowledge. All this information must be represented and managed efficiently to guarantee the feasibility of the Semantic Web. Querying and reasoning over instances of ontologies will make the Semantic Web useful.

Knowledge representation is a well-known problem for artificial intelligence researchers. Explicit semantics is defined by means of formal languages. Description logics (Nardi & Brachman, 2003) is a family of logical formalisms for representing and reasoning on complex classes of individuals (called concepts) and their relationships (expressed by binary relations called roles). DL formalism allows the description of concepts, relationships, and individuals (i.e., the knowledge base). All of them, together with complex concept formation and concept retrieval and realization, provide a query/reasoning language for the knowledge base. Research in description logics deals with new ways to query a knowledge base efficiently. On the other hand, knowledge representation research does not deal with large amounts of information, and its results can only be applied to very small knowledge bases (with a small number of instances), which are not the knowledge bases we expect to find on the Semantic Web. As a consequence, reasoning algorithms are not scalable and usually are main-memory-oriented algorithms.

All these problems increase when we deal with the context of the Semantic Web. In such an environment, reasoning and querying should be scalable, distributable, and efficient. In this context, we will find a fairly large amount of distributed instances. On the Semantic Web, not only reasoning on concepts is necessary, but also reasoning at the instance level and efficient instance retrieval. Therefore, it is necessary to allow (efficient) queries and (efficient) reasoning to be compatible in such a distributed (massive in instances) environment as the Semantic Web. This makes it necessary to develop efficient knowledge storage mechanisms that use secondary memory (in order to guarantee scale-up) and allow efficient reasoning about concepts and relationships defined by an ontology, and about its instances as well. Furthermore, it is fundamental to provide efficient disk-oriented reasoning mechanisms, particularly for the instance retrieval problem. We believe database technology (not necessarily database systems) is a must for this purpose, because it is developed to deal with large amounts of data and massive storage.

BACKGROUND

Description Logics

Description logics (DL) are a logical formalism, related to semantic networks and frame systems, for representing and reasoning on complex classes of individuals (called concepts) and their relationships (expressed by binary relations called roles). Typically, we distinguish between atomic (or primitive) concepts (and roles), and complex concepts defined by using DL constructors. Different DL languages vary in the set of constructors provided.

A DL knowledge base has two components: (1) a terminological part (the Tbox) that contains a set of concept descriptions and represents the general schema modeling the domain of interest and (2) an assertional part (the Abox) that is a partial instantiation of this schema,
comprising a set of assertions relating either individuals to classes or individuals to each other. Many of the applications only require reasoning in the Tbox, but in an environment like the Semantic Web, we also need Abox reasoning. Figure 1 shows an example of a description logic Tbox. Figure 2 shows an example of an Abox for this Tbox.

The reasoning tasks in a Tbox are consistency (satisfiability), which checks if knowledge is meaningful; subsumption, which checks whether all the individuals belonging to a concept (the subsumee) also belong to another concept (the subsumer); and equivalence, which checks if two classes denote the same set of instances (Nardi & Brachman, 2003). All of these reasoning mechanisms are reducible to satisfiability as long as we use a concept language closer under negation.

Typically, the basic reasoning tasks in an Abox are instance checking, which verifies whether a given individual is an instance of (belongs to) a specified concept; knowledge base consistency, which amounts to verifying whether every concept in the knowledge base admits at least one individual; and realization, which finds the most specific concept an individual object is an instance of (Nardi & Brachman, 2003).

In recent years significant advances have been made in the design of sound and complete algorithms for DLs. Moreover, systems using these algorithms have also been developed (Haarslev & Möller, 2001; Horrocks, 1998). Most of these approaches only deal with Tbox reasoning, but in an environment like the Semantic Web, we also need Abox reasoning. Although some DL systems provide sound and complete Abox reasoning, they provide very weak Abox query language. A query means retrieving instances that satisfy certain restrictions or qualifications and hence are of interest for a user.

**Web Ontology Languages**

The recognition of the key role that ontologies play in the future of the Web has led to the extension of Web markup languages in order to provide languages to define Web-based ontologies. Examples of these languages are XML Schema,1 RDF,2 and RDF Schema (Brickley & Guha, 2002).

Even with RDF Schema, RDF has very weak semantics. Still, it provides a good foundation for interchanging data and enabling true Semantic Web languages to be layered