Proc. of Database Systems for Advanced Applications (DASFAA'2007), Bangkok, Thailand, April 9-12, 2007.

Benchmarking data schemes of ontology based databases

Hondjack Dehainsala, Guy Pierra and Ladjel Bellatreche

Laboratory of Applied Computer Science (LISI) ENSMA - Poitiers University, France

Nowadays, ontologies are more and more used in several research and application domains, in particular e-commerce, e-business and the Semantic Web. Several tools for managing ontology data and ontology-based data (also called individuals or ontology class instances) are available. Usually, ontology-based data manipulated by these tools are stored in the main memory. Thus, for applications that manipulate a large amount of ontology-based data, it is difficult to ensure an acceptable performance. So, there is a real need for storing ontology-based data in database schemas to get benefit of the capabilities offered by DBMSs (query performance, data storage, transaction management, etc.). We call this kind of databases Ontology-Based Databases (OBDBs).

Over the last five years, two main OBDB structures for storing ontology and ontology-based data were proposed [1]. In the single table approach, the description of classes, properties and their instances are described by means of *triples* (subject, predicate, object) stored in a single table, called the *vertical table*. In the dual scheme approach, ontologies are described by specific schemas, depending upon the ontology model, but instances are stored either as set of *triples* in a single vertical table, or in a set of unary and binary tables, with one table per class, representing the identifiers of its instances, and one table per property, representing the pairs (instance identifiers, property value), this structure is known as the *decomposition model*. Unfortunately, all these approaches are poorly adapted when ontology-based data contains a large number of instances described by many properties values. In this case, any query requires a large number of join operations. This kind of ontology-based data is largely used in several application domains, particularly in e-commerce and e-engineering (which are our two main application domains). We have proposed a new architecture of OBDB, called OntoDB (Ontology Data Base). In this paper, we present the main characteristics of our approach, and the results of the benchmark used to compare our new structure with the existing structures. This benchmark uses a real ISO-standardized ontology for electronic commerce.

1 Overview of the OntoDB model

Our approach requires that ontology-based data fulfil three requirements: (1) each property is defined in the context of a class that specifies its *domain* of application, and it is associated with a *range*, (2) all the classes to which an instance belong have exactly one minimal class for the subsumption relationship, this class is the instance *base class* and (3) only properties that are *applicable* for

This work has been partly supported by the French ANR under grant ANR05RNTL02706 (e-Wok-Hub).

its base class may be used for describing an instance. Note that these requirements are fulfilled in a number of cases including e-commerce data. Then the OntoDB model consists of four parts [2]. The *ontology* and *meta-schema* parts are used for storing respectively ontologies data, and ontology models within a reflexive meta-model. The *meta-data* part is the usual database catalogue that describes the table structure of the three other parts. Finally, the *data* part is used for storing ontology-based data. Unlike in classical approaches, in OntoDB ontology class instances are stored using a *table per class* approach. It consists in associating a table to each ontology class. Columns of this table represent those rigid properties of the class that are associated with a value for at least one of its instance. Property values of each instance are represented in the same row.

2 Benchmarking OBDB models

Our benchmark is based on a real ontology standardized as IEC 61360-4. This ontology describes the various kinds of electronic components together with their characteristic properties. It is composed of 190 classes with a total of 1026 properties. We have generated automatically various sets of ontology-based data by varying the number of instances per classes and the number of properties values per instance. The size of test data falls within the range 0.3 GB-4 GB. Our test was based in three kinds of queries: (1) targeted class queries, where the user is supposed to know the root class of the subsumption tree to be queried; (2) non targeted class queries, where the user does not know what kind of ontology class she is looking for, and (3) insertion and update queries. Details of all our tests and results are presented in [3]. For queries (1) and (3), our table per class approach outperforms the two classical approaches as soon as more that one property value is requested. As a rule, the ratio is bigger than 10. The only case where the decomposition model is better than our approach is for the non targeted class queries when the user requests a very small number of property values. We note that this kind of queries nearly never happens in our application domain. Engineers always knows what they are looking for before searching for property values.

Our conclusion is that the OntoDB approaches outperforms all the classical approaches for processing in particular e-commerce data.

References

1. V. C. A. Magkanaraki, S. Alexaki and D. Plexousakis. Benchmarking rdf schemas for the semantic web. In *Inter. Semantic Web Conference (ISWC)*, 2002.
2. L. Bellatreche, G. Pierra, X. Nguyen, H. Dehainsala, and Y. Ait Ameur. An a priori approach for automatic integration of heterogeneous and autonomous databases. *Inter.Conf. on Database and Expert Systems Applications*, (475-485), 2004.
3. H. Dehainsala, G. Pierra, and L. Bellatreche. Managing instance data in ontology-based databases. Technical report, LISI-ENSMA,http://www.lisi.ensma.fr/ftp/pub/documents/reports/2006/2006-LISI-003-DEHAINSALA.pdf, 2006.