

USE OF DATABASES FOR STORING XML DOCUMENTS

Deise de Brum Saccol

Multimedia and Hypermedia Lab
Information Systems

Centro Universitário Luterano de Palmas – CEULP/ULBRA
Teotônio Segurado 1501 SUL – ZIP CODE: 77019-900
Palmas, TO, Brazil

Abstract: XML has been used as a suitable format to represent and exchange information, mainly due to its flexibility in the definition of structures (user responsibility) and the clear separation between data and schema related to these data. However, the fast growth of this kind of information, usually accessed through WEB, has caused not so efficient storage ways. This paper compares some existent storage ways of XML documents and presents a development project proposal of a XML Database Management System (DBMS).

Keywords: XML, data storage, database, semantic WEB, ontologies.

1 Introduction

The use of conventional databases (relational, object-oriented, relational object-oriented) has been relatively efficient to manage structured information. This highly typed information presents constant representation structure. Thus, *a priori* schema definition is suitable to be used by the data that will be inserted later in the database.

On the other hand, semistructured data do not present a fixed structure. The data schema, generally obtained *a posteriori*, results of data processing and search for standards that can be used in the definition of a common structure. Typical examples of semistructured data are the ones found in the WEB or obtained through integration of information [1]. Since XML does not have a rigid structure in data representation, it can be considered as a format for semistructured data.

The absence of a fixed structure in semistructured data makes impracticable the use of many techniques adopted by the conventional DBMS's. In this way, some adaptations are need to be done between the two formats (conventional *versus* semistructured) so that it is possible the storage of XML data in a structured format.

The goal of this paper is to present the main existing ways for XML documents storage, emphasizing the use of specific DBMS's for this kind of information. The paper is organized as follows: section 2 presents the main forms of XML documents storage; in section 3, two DBMS's specifically built for the storage of semistructured data are shown. Section 4 presents the *OntoDB*, the DBMS proposal of this paper; final considerations are presented in section 5.

2 XML documents storage

There are four possible approaches to store semistructured data [3]. The first one consists in using text files. This technique can be used when it is just wanted to keep the information of semistructured data, without the use of a management software for efficient storage and access to the information.

The second approach consists in using a relational database system. In this case, XML data are mapped to tables of a relational schema and the queries posed over the semistructured data need to be translated into SQL queries.

The third approach consists in using a object-oriented database system. In this case, the wide data modeling capabilities of this kind of system are explored. An implemented commercial system in this category is *O2* [2]. The *Monet* project also follows this approach [10].

The latest alternative is to construct a specific purpose database system. Examples of this category are *Lore* [5] and *Tamino* [7]. These systems are specifically built to store and access semistructured data, using special structures and indexes and specific techniques of query optimization.

We can point out some considerations about the four presented approaches. The use of a *.xml* text file for data storage can be efficient when the kind of access to these information is simple and static. However, data

from WEB usually present a dynamic and evolutionary behavior, coming from databases with support to concurrent access and crash recovery. Relational databases systems present good scalability and are widely used to manage conventional data. However, for having a fixed schema, many columns could be empty with the insertion of XML data, since it is necessary to adapt the structural flexibility of XML data to the inflexibility of the relational schema. In the same way, the current generation of object-oriented databases is not mature enough for efficient query processing in very large databases.

Theoretically, specific purpose systems should have good acceptance and seem the best alternative. However, a long time must be spent so that such systems get mature and reach good scalability for large number of data. In this direction, some commercial products already exist, as briefly described in next section.

3 XML DBMS's

Amongst some proposals of XML DBMS's, two systems are distinguished: *Lore* [5] and *Tamino* [7]. These databases support integrated information from heterogeneous data sources, being able to search existing information in external sources.

Lore uses a data model based in graphs, called *OEM* (Object Exchange Model). Element tags are represented as *edges*. Each existing node in the graph is an object. The used language, LOREL, is an extension of OQL language (Object Query Language). The language

accesses and updates data, realizes coercion of types and uses path expressions. Still, it supports insertion and removal of edges, creation of new nodes (objects), modification of atomic values, aggregation and grouping functions. The Lore architecture uses an interface to access the system through applications or a API (Application Program Interface).

Tamino uses the XML document structure as data model, being capable to store and process XML data without the need to convert for other formats (native storage). Tamino uses the X-Query language, based on XPath specification (proposed by W3C [8]). XML document is structured as a tree, with different types of nodes. The X-Query language is based on path expressions. The Tamino manager is a graphical interface where the administrator can create, remove and modify XML databases located in the server.

The main goal of these DBMS's is to provide a practical and efficient storage for querying and updating semistructured data. Lore has not become a commercial product yet. Tamino is already commercialized; it is more complete and offers more resources for the user, mainly related to the graphic interface used to access the system.

4 Proposal – *OntoDB* DBMS

The construction of the XML DBMS considered in this work (*OntoDB*) is in architecture definition and modeling phase. The proposed architecture is shown in Figure 1:

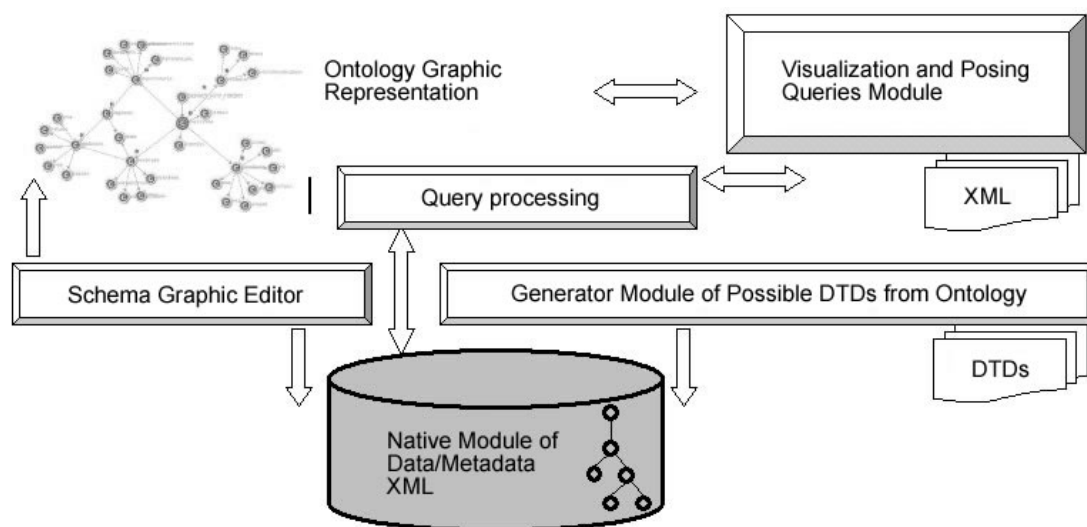


Figure 1 Generic Architecture - *OntoDB*

Since a set of XML documents concerning to the the same domain problem can be validated against distinct DTDs (Document type definition), the considered DBMS uses ontologies to represent structures of the stored data. According to [9], an ontology defines the used terms to describe and to represent the knowledge. Ontologies are used for people, databases and applications that need to share domain information, including definitions of basic concepts and the relationships between them.

The main system modules are described below:

- *Schema Graphic Editor*: this module is responsible for the ontology construction, which will work as a generic structure for the Generator Module of possible DTD's. The constructed tool generates the internal representation of the ontology, with all the associated metadata. The ontology is constructed from a meta ontology.

- *Generator Module of DTD's*: this module generates all the possible DTD's that could be used as validation structure for the stored XML documents. XML documents must belong to the problem domain described by the ontology.

- *Visualization and Posing Queries Module*: through this module, the user poses (graphically) queries based on the ontology. This query is sent to the Query Processor, which access the data repository, processes and returns the result to the user as XML documents.

- *Query Processing*: this module implements the grammar parser that defines the language of OntoDB system. The proposed language, OntoQuery, is based on path expressions defined from the ontology.

The architecture is in initial implementation. Thus, it does not propose a queries optimizer. This module will be introduced in a posterior phase. The code is being implemented in Java. The definition of the grammar for the OntoQuery language was made using JCC (Java Compiler Compiler), parser that reads the specification of the grammar and converts it for a Java program that recognizes queries in accordance with this grammar.

5 Final Considerations

The absence of efficient ways for XML documents storage has encouraged the research of new technologies. Some existing commercial products, as Oracle 9i DBMS, has added some interesting features

for management of this kind of information (for instance, the *XMLType* data type) [6]. Lacking of mature solutions, the construction of specific XML databases from scratch appears as a good alternative.

References

- [1] ABITEBOUL, S. et al. "Querying Semi-Structured Data". In: INTERNATIONAL CONFERENCE ON DATABASE THEORY, ICDT, 6., 1997, Delphi, GR. Database Theory: proceedings. Berlin: Springer-Verlag, 1997. Available at: <<http://citeseer.nj.nec.com/abiteboul97querying.html>>.
- [2] CHRISTOPHIDES, V. et al. From structured documents to novel query facilities. SIGMOD Record, New York, v.23, n.2, p. 313-324, June. 1994.
- [3] FLORESCU, D. et al. A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database. 1999. (Technical Report, n. 3680). Available at: <<http://rodin.inria.fr/Epubsbyyear.html>>.
- [4] GRAVES, M. Designing XML Databases, PH PTR, New Jersey, EUA, 2002.
- [5] McHUGH, J. et al. Lore: A Database Management System for Semistructured Data. Available at: <<http://www-db.stanford.edu/lore/pubs/lore97.pdf>>. SIGMOD Record, New York, v.26, n.3, p. 54-66, Sept. 1997.
- [6] ORACLE WEB SITE. Oracle 9i Database - The New XMLType Datatype. Available at: <<http://otn.oracle.com/products/oracle9i/daily/Jul06.html>>.
- [7] SOFTWARE AG WEB SITE. TAMINO XML SERVER. Available at: <<http://www.softwareag.com/tamino>>.
- [8] WORLD WIDE WEB CONSORTIUM WEB SITE. XML Path Language (XPath) Version 1.0-W3C Recommendation 16 November 1999. Available at: <<http://www.w3.org/TR/xpath>>.
- [9] WORLD WIDE WEB CONSORTIUM WEB SITE. Requirements for a Web Ontology Language - W3C Working Draft 07 March 2002. Available at: <<http://www.w3.org/TR/2002/WD-webont-req-20020307/>>.
- [10] ZWOL, R. et al. Modelling and querying semistructured data with MOA. In: WORKSHOP ON SEMI-STRUCTURED DATA AND NONSTANDARD DATA FORMATS, 1999, Jerusalem, IS. Proceedings... Jerusalem: VLDB Endowment, 1999.