

BDOviedo3: Data XML Storage and Management

Ana Belén Martínez, Darío Álvarez, Francisco Ortín, Juan Manuel Cueva,
and M^a Ángeles Díaz

Dept. Informatics. University of Oviedo. Calvo Sotelo s/n 33007, Oviedo, Spain
{belen,darioa,ortin,cueva,fondon}@lsi.uniovi.es

Abstract. OOBDBMS provide a good solution for the management of XML documents. This article shows the different approach (specific and generic) of coupling an object model with XML and emphasizes the possibilities offered by BDOviedo3 system. It is an object-oriented database built over an integral object-oriented system. The flexibility provided by its object-oriented construction and the extensibility of its mechanisms (such as indexing) can transform it into a XML document manager.

1 Introduction

The XML language (eXtensible Markup Language) is now a standard for information interchange. Due to the high amount of data involved, aspects related to the management of XML documents, such as efficient storage, indexing, queries, transactions, data integrity, multi-user access, etc., traditionally provided by databases, have special relevance.

This paper presents the options provided by Object-Oriented Database Management Systems (OODBMS) for the management of XML documents, with an emphasis on the possibilities offered by BDOviedo3 [9]. BDOviedo3 is an OODBMS built on a persistent object-oriented abstract machine [2].

Section 2 introduces an overview of the main alternatives that can be used for the management of XML documents using OODBMS. Next section presents the BDOviedo3 system. Finally, in section 4, a number of extensions of BDOviedo3 system for the storage and management of XML documents are discussed.

2 Object-Oriented Database Systems and XML

The potential of OODBMS for the management of XML documents is well known [6]. Besides, they have a number of advantages when compared with other models, such as the relational model [1].

The goal of coupling an object model with XML has two main approaches: *specific* and *generic* [5]. *Specific approach* is very adequate for XML documents where data is what is really interesting. Describes a new class for each new kind of information. It models the data in the XML document as a tree of objects that are specific to the data in the document. There are already a number of products implementing XML data

binding [3,7] and even many of these can transfer data between the objects and the database as well. *Generic approach* is appropriate in general for *document-centric* applications. The Document Object Model (DOM) uses such as approach. Generic approach considers all document information (tags, attributes, etc.) as objects of pre-defined classes, interconnected between each other with links that preserve XML structure. This approach is used by several model-based native XML databases [10,16]. A model-based native XML is a database that stores a binary model of the document (DOM or a variant) in an existing or custom data store.

3 BDOviedo3

BDOviedo3 is an OODBMS built over an integral object-oriented (OO) system [9], which provides direct support for the OO paradigm [2]. This integral system is supported by an OO abstract machine that provides a set of minimal capabilities (persistence, concurrency, security and distribution), and an OO operating system that extends them. The abstract machine provides a common object model shared for the other system elements.

The main element of the architecture of the BDOviedo3 OODBMS is the engine [11], that provides the basic functionality of the OODBMS. It is structured into eight managers (extensions, schemas, integrity, queries, indexes, transactions, authorization, and distribution). As an example of the facilities provided by the integral system, the extensibility [14] of the indexing mechanism can be mentioned. This mechanism easily allows the incorporation of new indexing techniques. These techniques are implemented as object sets with a particular interface. Every new indexing technique added to the system, by using inheritance, will have that interface. These classes can be used and extended by any user, thus reusing the OODBMS code itself. The initial set of indexing techniques coded are Single Class, CH-Tree and Path Index [15].

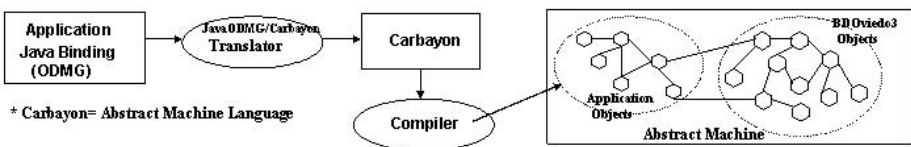


Fig. 1. Architecture of BDOviedo3

The languages selected [9] for the manipulation and querying of the database are the ones proposed by the ODMG standard [4]: Definition Language (ODL), Management Language (OML), and Query Language (OQL) in its Java binding incarnation. However, the addition of new languages to the OODBMS just involves the coding of a translator of the language into the bytecode of the abstract machine using the already present functionalities of the objects of the database engine.

4 Possibilities for the Management of XML Offered by BDOviedo3

BDOviedo3, as mentioned before, is an OODBMS. As such, it can host the management of XML documents. Nevertheless, the special features of the system offer a big number of possibilities. Some of them are discussed below.

The first one is along the lines proposed in [8], for instance. It is based on extending the ODMG standard to support structured and semi-structured data. This implies the existence of a model to represent semi-structured data (OEM in the case of Ozone [8]). The extension of the OQL language to support queries on hybrid data is also required. The solution in BDOviedo3 will need the addition of a new layer to deal with these extensions. First, a new schema manager is needed. This schema manager will add new types representing semi-structured data to the ODMG class declarations. Then a new preprocessor translating extended OQL queries into plain OQL should also be added.

In order to achieve a *generic* management of XML documents, a solution will be the developing of a persistent DOM (PDOM style [5]). However, in BDOviedo3, the implementation will be very easy, as all the aspects related to the synchronization of objects between main memory and hard disk, object lifecycle control, etc. are transparent to the programmer of the persistent DOM. A module for making XML specific queries (XQuery for instance) can also be easily added. The extensibility of the indexing mechanism allows to introduce without effort new indexing techniques that prove to be better [12,13] for searches involving these documents.

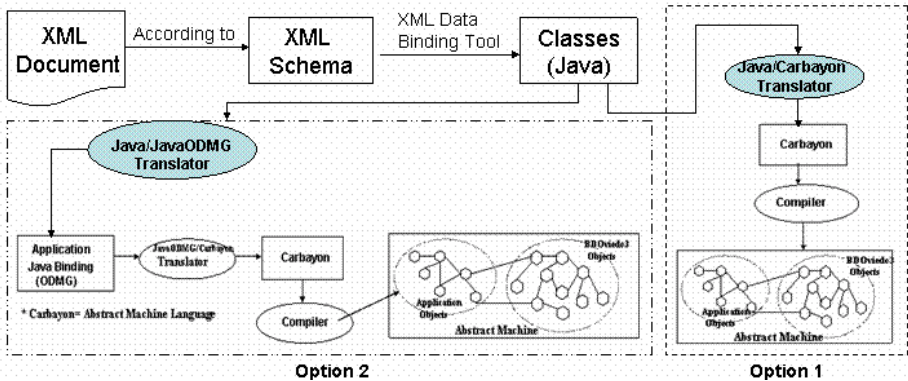


Fig. 2. Alternatives to add specific XML document management to BDOviedo3

Finally, in order to achieve a *specific* management of XML documents there are a number of possibilities. The BDOviedo3 languages are ODMG standard (Java binding), so any of the Java language tools implementing the XML data binding (such as [3,7]) can be directly used to translate between the database schema and the XML document schema. Then we have two options (Fig.2). The first one is to translate the Java source generated by the tool to the language of the abstract machine. Then the

schema is now persistent (thanks to the persistence of the machine), and data in main memory is synchronized with data stored in the database. The other option is to translate into a language understood by the ODBMS (the ODMG extended Java), which then in turn is translated to machine bytecode by an already existing tool in the ODMS. An extra step is taken but the translator is simpler.

5 Conclusions

Object-Oriented databases are a feasible and adequate option for the storage and management of XML documents. The BDOviedo3 OODBMS offers different possibilities for the storage and management of XML documents, for data-centric and document-centric applications as well. The flexibility provided by its object-oriented construction over a persistent OO abstract machine, and the extensibility of its managers and mechanisms (such as the indexing mechanism) can easily transform BDOviedo3 into a XML document manager.

References

1. Abiteboul, S., Buneman, P., Suciú, D. Data on the Web. From Relations to Semistructured Data and XML. Morgan Kaufmann Publishers (2000)
2. Álvarez D. et al. An Object-Oriented Abstract Machine as the Substrate for an Object-Oriented Operating System. 11th ECOOP'97. Finland (1997)
3. Castor. <http://www.castor.org/>. January (2003)
4. Cattell, R., Barry, D., Berler, M. et al. The Object Data Standard: ODMG 3.0. Morgan Kaufmann Publishers (1999)
5. Chaudhri, A. and Zicari, R. Succeeding with Object Databases. John Wiley & Sons (2001)
6. Fattersack, P., C. Espert, and Bolf, D. Good Performances for an SGML Object Database System. Proceedings of the SGML 97 Conference. Boston (1997)
7. Java Architecture for XML Binding (JAXB). Proposed Final V.0.90. (December 2002) <http://java.sun.com/xml/jaxb/>
8. Lahiri, T., Abiteboul, S. and Widom, J. Ozone: Integrating Structured and Semistructured Data. 7th Workshop on Database Programming Languages, Scotland (1999)
9. Martínez, A.B. Un Sistema de Gestión de Bases de Datos Orientadas a Objetos sobre una Máquina Abstracta Persistente. PhD thesis. University of Oviedo, ISBN 84-8317-277-1 (2002)
10. MindSuite 2.0 XDB. <http://xdb.wiredminds.com/> (2003)
11. Ortín, F., Martínez, A.B. et al. A Reflective Persistence Middleware over an Object Oriented Database Engine. 14th Brazilian Symposium on Databases. Brasil (1999)
12. Rizzolo, F. and Mendelzon, A. Indexing XML Data with ToXin. 4th International Workshop on the Web and Databases (with ACM SIGMOD). Santa Barbara (2001)
13. Sacks-Davis et al. Indexing Documents for Queries on Structure, Content and Attributes. International Symposium on Digital Media Information Base, Nara, Japan (1997)
14. Stonebraker, M.R. Inclusion of New Types in Relational Data Bases Systems. Proc. 2nd IEEE Data Engineering Conf. Los Angeles (1986)
15. Yu, C. and Meng, W. Principles of Database Query Processing for Advanced Applications. Morgan Kaufmann (1998)
16. 4Suite. <http://4suite.org/index.xhtml> (2003)