

# Design and Implementation of XML DBMS Based on Generic Data Model\*

JongSeon Lim<sup>1</sup>, Sung yoon Bang<sup>2</sup>, and Kyung-Soo Joo<sup>1</sup>

<sup>1</sup>Dept.of Computer Science and Engineering, College of Engineering SoonChunHyang Uni.,  
P.O Box 97, Asan, ChungNam, Korea, 336-745  
[ronmer@chol.com](mailto:ronmer@chol.com), [gsoojoo@sch.ac.kr](mailto:gsoojoo@sch.ac.kr)

<sup>2</sup>Hanseo University Computer Center, SeoSan, ChungNam, Korea, 356-706  
[sybang@hanseo.ac.kr](mailto:sybang@hanseo.ac.kr)

**Abstract.** Nowadays XML is used for exchanging information in e-Commerce, especially B2B. Necessity of XML DBMS has being increased to efficiently process XML data. So a lots of database products for supporting XML are rapidly appeared in the market. In this paper, we made an XML DBMS system based on Generic Data Model. First we developed XML Adaptor based on Generic Data Model and added it on relational DBMS for developing XML DBMS. XML Adaptor is composed of Query Converter and XML Repository System. The Query Converter parse commands that are for XML data manipulation and then call the relevant component of XML Repository System for relational database operation. The XML Repository System handles relational database operations such as create, delete, store, and etc. In this way we can use a relational DBMS for manipulation XML data. Therefore we can build more economically XML DBMS.

## 1 Introduction

In order to support XML data, current mainstream relational DBMS uses tools to convert other data formats to and from XML, which causes slow-down in DBMS processing speed and other problems. Thus we are likely to see increased demands for XML DBMS that can store or load XML data in its original form. Also, since there are many customers who want single DBMS platform, relational DBMS with XML support is likely to become general trend rather than pure XML DBMS. Relational DBMS currently takes up more than half the current market and they have significant difficulties in transforming to the new XML DBMS. In order to overcome these limitations, enterprises can employ two kinds of method of either converting the relational DBMS to XML DBMS or combining relational DBMS and XML together. For the former, DBMS itself is transformed inside and out and costs a great amount. But all the powerful features of XML DBMS in XML document processing. For the latter, relational DBMS adapts support for XML, allowing for easy storing and

---

\* This work was supported by the University IT Research Supporting Program under the Ministry of Information Communication of Korea.

managing of XML documents. However, it will not be able to utilize all the powerful XML document processing features the former has.

In this research an XML DBMS is designed on top of a relational DBMS as an XML adaptor. Causing no modifications on the existing relational DBMS, it will also be able to perform as an XML DBMS. This can bring significant benefits in system implementation with least expenses. Section 2 describes relevant researches and technologies, section 3 XML DBMS design, section 4 XML DBMS implementation and lastly in section 5, we will draw conclusion for this research.

## 2 Related Works

### 2.1 EJB

J2EE (The Java 2 Platform, Enterprise Edition) is the standard for developing multi-tier enterprise applications. In other words, J2EE simplifies the enterprise application developing process by building on top of standardized and modular components, a multi-tier enterprise application, providing full service to the respective component and automatic application processing without complicated codes. J2EE is a platform for enterprise solutions that features simplicity, migration, extension, integration and such. Further, J2EE can be said to be a set consisting of JSP, Servlet, Java Bean, EJB, and so forth, of which EJB is the core technology[3].

EJB is a component based distributed object technology that possesses the crucial core technology responsible for abstract data and business logic in enterprise level application development. Moreover, EJB container helps the developer by providing built-in services such as automatic enterprise bean life-cycle management, state information management, security, transaction processing, and continuous processing, so that the developer only needs to be concerned with the business logic.

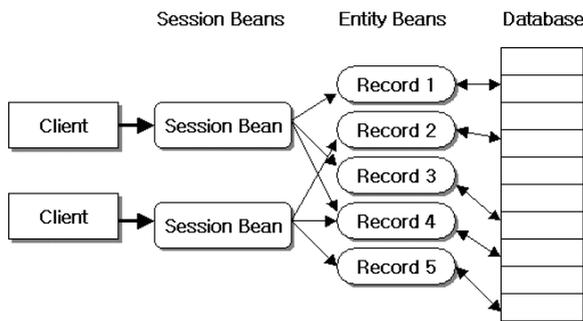


Fig. 1. Session beans and Entity Beans.

Enterprise Bean is an EJB component that clients can call and use. Enterprise Bean can be developed either as Session Bean and Entity Bean. Fig. 1. shows the relationship between the Session Bean and Entity Bean[4][5].

## 2.2 XML Database

An XML database is a collection of XML documents that persist and can be manipulated. Historically, documents were developed for communication between humans. With the advent of computers, documents may also be used for computer-computer communication, human-computer communication, or computer-human communication. Each of these communication paradigms has its own requirements, such as aesthetics, tolerance for ambiguity, preciseness, and flexibility[1][2].

XML documents tend to be either document-processing-oriented or data-processing-oriented. Document-processing-oriented documents are those in which XML is used for its ability to capture natural (human) languages, such as in user's manuals, static Web pages, and marketing brochures. They are characterized by complex or irregular structure and mixed content and their physical structure is important. The processing of the document is focused on the final presentation of the information to the user; thus, they may be called presentation-oriented documents. Data-processing-oriented documents are those where XML is primarily for data transport. These include sales orders, patient records, and scientific data. The physical structure of data processing-oriented documents, such as the order of elements or whether data is stored in attributes or sub elements, is often unimportant. They are characterized by highly-regular structure with many repetitions of those data structures. The processing of the document is usually focused on its use and exchange by applications; thus, they may be called message-oriented documents. Document-processing-oriented and data-processing-oriented documents are sometimes called document-oriented documents versus data-oriented documents, respectively, but those terms are ambiguous as every document always contains data[2].

## 2.3 Generic Data Model

The Generic Data Model is derived from the previous data models and is used for the storage mechanisms. It provides a basic data model for a variety of applications that can also be extended using more of the ideas in the previous data models [2].

### 2.3.1 Operations

#### 1) Database

The operations on the Database type are made explicit in the Generic Data Model. Table 1. are representation to database operations in Generic Data Models, and abbreviated to arguments and return values[2].

#### 2) Document

Two operations are added to the other Data Model. "GenerateDocument", generates the document next as suggested in the W3C Data Model and a new operation "selectElement", operation is more general than the "getElementByTagName" of the Node-centric Data Model.

**Table 1.** Database operations.

Database Operation	Description
newDatabase()	Create a new database.
newDocument()	Create a new document.
storeDocument()	Store a new document, return the new document.
retrieveDocumentName()	Retrieve from database.
deleteDocument()	Delete from database the document, return true if the document was originally there.
getAllDocuments()	Retrieve from database a list of all its documents.
selectDocument()	Select from database all documents, which satisfy constraint.

### 3) Element

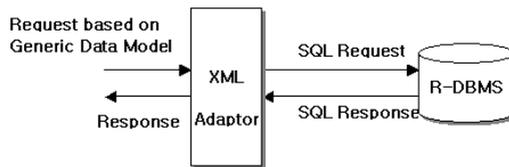
The twelve operations for Element are the same as the Simple Data Model. The operations are taken from the Simple Data Model rather than Node-centric Data Model, because the DOM Node operations are not likely to be efficient for large databases.

Operations in general data model for XML elements are shown above. There are lots more operations other than the ones shown, but operations for database, attributes, and elements are introduced here. The ones proposed in this research also applies database operations from general data model.

## 3 XML DBMS Design

### 3.1 XML DBMS Structure

XML DBMS is created using the existing relational DBMS as its infrastructure and adding XML adaptor. Fig. 2. shows the system-wide structure. Users can make queries through general data model and XML adaptor without regards of the DBMS. XML adaptor can then store, search, and delete XML documents stored in the database through SQL.



**Fig. 2.** XML Adaptor.

Fig. 3. is a more detailed look at the XML adaptor in Fig. 2. Queries are basically processed through general data model query language and query converter converts the general data model based queries into SQL. Further, these converted queries can use the storage management system that are built on EJB to perform store, search, delete and other tasks. In order to perform various operations in general data model, a few additional components have been added.

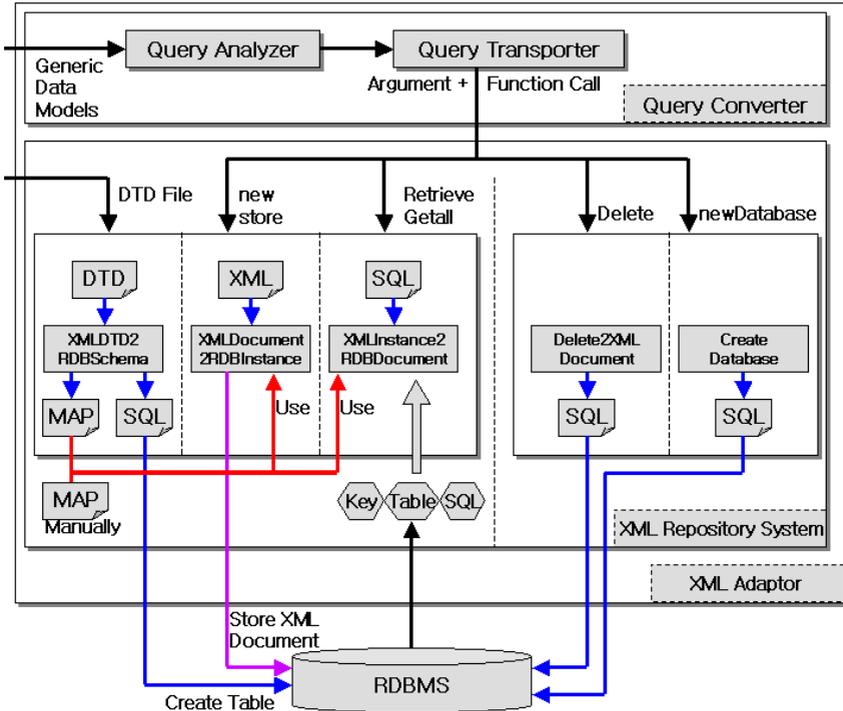


Fig. 3. Detail of XML Adaptor.

Query analyzer analyzes the query and extracts the operation part and argument, return value parts. These 3 extracted parts are then handed over to the query converter. Query converter in turn applies the operation and argument parts to convert them to the most appropriate commands or SQL and turns it over to the repository system. Repository system consists of components that embody every database operation for generic data model, and queries the database using the component called by the query analyzer.

### 3.2 Query Converter

Query converter consists of query analyzer and query converter. Query analyzer performs the task of analyzing generic data model based query and transporting the

result values to the query converter. General generic data model query consists of operation, argument, and return value as shown in Fig. 5.

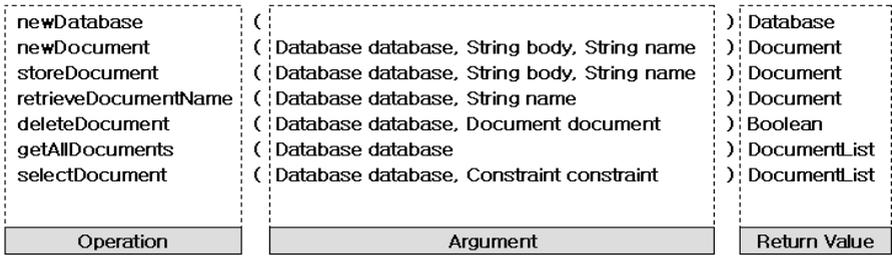


Fig. 4. Query structure of Generic Data Models.

Query converter is responsible for converting the query from the analyzer into XML repository system commands or SQL. Fig. 5. describes what format the query from Fig. 4. is converted into. Fig. 5. shows the decision process on how generic data model database operations are converted. Component to call is decided by the operation, argument, and return value. Then database name and document name are determined appropriately for the respective component.

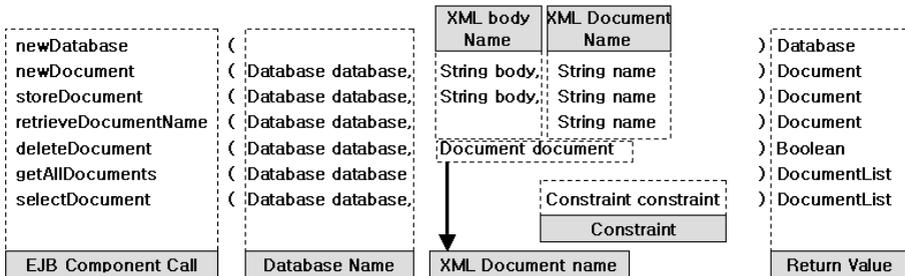


Fig. 5. Arguments of Generic Data Models.

Each operation for generic data model takes different arguments, so by separating each respective argument and calling the appropriate operation, we can store, search, and delete data through the repository system as shown in Fig. 5.

### 3.3 XML Repository System

XML repository system is divided into conversion, storage, search, deletion and database generation modules. Conversion module takes XML DTD as the input and converts them to relational database schemes. Storage module stores the XML documents in the relational database. Search module searches the relational database with key, table, and SQL, then documents the search results into XML. Deletion module looks up the table for a single XML document and deletes it. Database generation module generates a database. The structure of XML repository system is

designed by making the XML DBMS features into EJB components and implementing the system by assembling each of these EJB components.

### 3.3.1 XML Repository System Functions

XML repository system functions are listed below:

- ① Conversion: Provides two methods for converting XML DTD into relational database scheme. First method takes the XML DTD as its input and automatically extracts the relational database scheme. Second method is a manual conversion method where designer provides a map that displays how XML DTD and relational database scheme interrelates and based on this mapping, relational database scheme is extracted from the XML DTD.
- ② Storage: Stores an XML document in the relational database according to the supplied mapping.
- ③ Search: 3 different methods can be used. First method uses the data in the root table to search every table and generates an XML document in the mapping order. Second method searches data in a single table to generate an XML document in the mapping order. Last method generates an XML document in the mapping order by accepting SELECT command from the user for data search.
- ④ Deletion : Deletes an XML document in the database.
- ⑤ Database Generation : Generates a database.

### 3.3.2 Relationship between XML Repository System and EJB Components

Components used for implementing XML repository system are shown in Fig. 6. An EJB component, 'XMLDTD2RDBSchema' is used for conversion, 'XMLDocument2RDBInstance' for storage, 'RDBInstance2XMLDocument' for deletion, and 'CreateDatabase' for creating a database. All these components are assembled together to implement XML repository system.

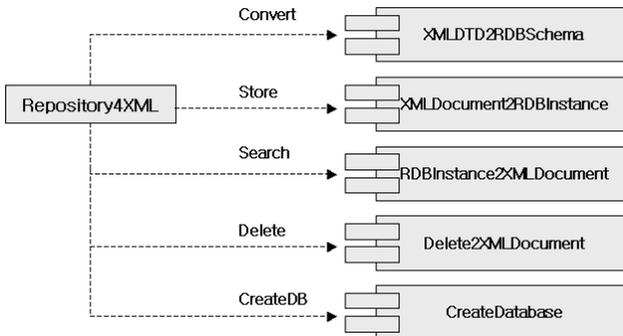


Fig. 6. EJB components of XML Repository System

## 4 Implement of XML DBMS

Windows 2000 Server, JBuilder 4.0, and Oracle 8i were used in implementing our system. Fig. 7. is a query for storing a new XML document in the database by using generic data model. The called component uses a storage component shown in Fig. 9. to store the XML document in the database. Similarly, Fig. 8. and 10 shows the conversion and search components that are called upon.

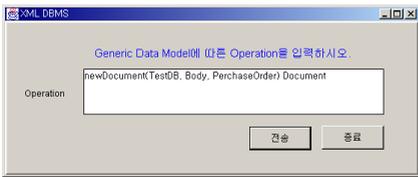


Fig. 7. “Newdocument” operation

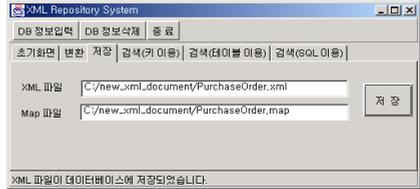


Fig. 8. Store XML document



Fig. 9. Convert XML document

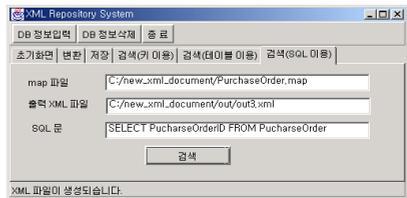


Fig. 10. Search XML document.

## 5 Conclusion

This research used existing relational DBMS as the infrastructure and XML DBMS was developed by adding XML adaptor for handling general data model based XML document. XML adaptor is composed of i) a converter that parses generic data model type command and calls appropriate component, and ii) XML repository system that performs the actual tasks on the relational database. XML repository system is built by assembling 5 components (i.e. conversion, storage, search, deletion, and database generation) that are based on EJB.

Using XML adaptor introduced in this research, XML data handling is made possible without having to replace the existing relational DBMS, ultimately lowering the cost for implementing an XML DBMS and enabling the usage of low-cost relational DBMS.

## References

1. Harvey Deitel, XML How to Program, Prentice Hall, 2002
2. Mark Graves, Designing XML Databases, Prentice Hall PTR, 2002
3. Vlada Matena; Beth Stearns, Applying Enterprise JavaBeans™ Component-Based Development for the J2EE™ Platform, Sun, 2001
4. Lee Jung-soo, Jung Sang-hyuk, Joo Kyung-soo, "A Design and Implementation of XML Repository System based on EJB component ", Korean Society for Internet Information
5. Lee Jung-soo, Joo Kyung-soo, "Designing Components for mapping from XML DTD to ORDB Schemas", Journal of Information Technology Applications & management, Vol.9 No.1 March 2002, pp72-83

**Jong-Seon Lim** received his B.S. degree from Dept. of Computer Science, Korea Uni. in 1997 and obtained M.S. degree from Dept. of Computer Science, Soonchunhyang Uni. in Korea.

Dept. of Computer Science, Graduate school Soonchunhyang Uni.

E-mail : [ronmer@chol.com](mailto:ronmer@chol.com)

**Sung yoon Bang** received his B.S. degree from Dept. of Computer Science, Korea National Open Uni. In 1996 and obtained M.S. degree from Dept. of Computer Science, Hoseo Uni. In 1998 and obtained Ph.D degree from Dept. of Computer Science, College of Engineering Soonchunhyang Uni. In 2003.

Hanseu University Computer Center manager

E-mail : [sybang@hanseo.ac.kr](mailto:sybang@hanseo.ac.kr)

**Joo Kyung-Soo** received his B.S. degree from Dept. of Mathematics, Korea Uni. In 1980 and obtained M.S. degree from Dept. of Computer Science, Korea Uni. In 1985 and obtained Ph.D degree from Dept. of Computer Science, Korea Uni. In 1993

Dept. of Computer Science, College of Engineering Soochunhyang Uni. Prof.

E-mail : [gsoojoo@sch.ac.kr](mailto:gsoojoo@sch.ac.kr)