Journal of Computer Science & Its Applications, December 2006, Vol. 13 No.1

Interoperability of Information Systems and Heterogenous Databases Using XML

Ayo, C. K. and Adebiyi, A. A.
Department of Computer and Information Sciences
Covenant University, Ota, Ogun State.
ckayome@yahoo.com, ariyo_adebiyi@yahoo.com

ABSTRACT

Interoperability of information systems is the most critical issue facing businesses that need to access information from multiple information systems on different environments and diverse platforms. Interoperability has been a basic requirement for the modern information systems in a competitive and volatile business environment, particularly with the advent of distributed network system and the growing relevance of inter-network communications. Our objective in this paper is to develop a comprehensive framework to facilitate interoperability among distributed and heterogeneous information systems and to develop prototype software to validate the application of XML in interoperability of information systems and databases.

Keywords: ***Interoperability, Information Systems, XML, Databases***

1. INTRODUCTION

Information has become the most critical resource in many organizations, and the rapid growth of networking and database technologies has had a major impact on information processing requirements. There is greater need for access to information for enhanced productivity, better information sharing and distribution. As a result, an increasing number of databases in different sites are being interconnected. In order to reconcile the contrasting requirements of the different database management systems (DBMSs), tools that enable users of one system to use another system's data and information are being developed. Efficient solutions for interconnecting and administering different database systems are also being investigated [1].

The wave of corporate mergers and growth of business on the Internet have boosted enterprise systems integration's profile in both Information Technology (IT) and business. Interoperability has been a basic requirement for the modern information systems' environment in the recent time. Data and information interoperability have gained increasing attention for several reasons: (1) excellent progress in interconnection afforded by the Internet, Web and distributed computing infrastructures, leading to easy access to a large number of independently created and managed information sources of broad varieties, (2) increasing specialization of work, but increasing need to reuse and analyze data, leading to creation of information and knowledge in the industrial context, in which enterprises must react quickly to the market changes [2].

In order to face this problem, enterprises must collaborate together. This implies at one hand high communication between their information systems and

at the other hand the compatibility of their practices. An important work of change must be done for practices standardization and harmonization [3]. This is the concept of interoperability. Interoperability can be defined as the ability of two or more systems or components to exchange information and to use the information that has been exchanged [4,5,6].

Although research on interconnecting different DBMSs has been under way for over a decade, only recently have many of the difficult problems been addressed. Through the evolution of the three-tier approach to client/server, the capability of integrating DBMSs has improved significantly. The traditional two-tier client/server approach included the layers of Client and Server. For small systems, the two-tier approach works reasonably well. For larger systems with greater number of connected clients and servers, and greater levels of complexity and requirements for security, there is a substantial need for three-tier architectures. Two-tier systems are notorious for their development of the "fat client", where excessive amounts of code running business logic are required to be loaded onto the client machine.

The three-tier approach breaks client/server components into the layers of Client (presentation layer), Middleware (business logic), and Server (data and resource management). The result is more efficient use of resources, and greater plug and play capabilities for both clients and servers. Clients can be superthin browser running JAVA applets, and server can be efficiently integrated and load-balanced [1].

With the advent of web servers, the three-tier model becomes n-tier since web server is often placed between the client and middleware layers. Hence, the issue of interoperability of heterogeneous information system and databases is of a major concern since solutions such as Electronic Data Interchange (EDI) and Enterprise Resource Planning (ERP) focus only on providing software for automating operations within tightly coupled organizations. For an organization to achieve full benefits from electronic commerce, a solution must automate the operations between trading partners.

An integration solution must cope with the complexity of integrating information from a number of varied information systems, spanning the entire length of the E-commerce continuum. A solution to provide a message format must be open and flexible enough for different applications to understand, process and respond to it.

Some developers are looking toward XML (e**X**tensible **M**arkup **L**anguage) to solve the problem of business-to-business integration. XML is the emerging standard that promises to bridge the communication gap between enterprise resource planning, electronic data interchange, and Web-based systems. Its real significance may emerge as a means of making it easier to create, deploy, and manage integration solutions over the Internet [7].

2. STATEMENT OF PROBLEM

The recent issues of bank recapitalization and subsequent mergers and acquisitions depict the importance of interoperability of information systems and databases. Following the consolidation cycle, the now bigger banks recognized that it had acquired dozens of information systems that run on diverse platforms, used different databases, and had unique user interface and communication features. This poses a serious challenge in the banking industry.

3. OBJECTIVES

i. To design a comprehensive system architecture for database interoperability.

ii. To show that XML is a veritable tool for interoperability of information systems and databases over existing tools.

iii. To develop a prototype software for interoperability of databases.

4. ANALYSIS OF EXISTING INTEROPERABILITY TOOLS

The following are existing tools of interoperability of information systems

1. CORBA (Common Object Request Broker Architecture)
2. DCOM (Distributed Component Object Model)
3. RMI (Remote Method Invocation)

4.1 CORBA

Sequel to the shortcomings of the client/server paradigm, there was a formation of a group known as Object Management Group (OMG), a consortium of nine companies that formed a distributed computing interoperability standard known as CORBA in 1989. The purpose of the OMG and its CORBA standard was to define a component-based software interoperability and communication standard. This standard was based on software development that models the real world through a representation of objects where these objects are the encapsulation of attributes, relationships and methods of software identifiable program components [8].

WEAKNESES OF CORBA

1. OMG produced CORBA products in the term of standard but not specification; this led to a situation where the CORBA standard product from one vendor would not work with the CORBA standard product from another vendor. The lack of a specification meant that there was no single test suite or compliance board that would verify that a given CORBA implementation actually met the CORBA standard.
2. CORBA usually requires Object Request Brokers (ORB) that are known to work together.
3. CORBA did not take into consideration the Internet and its coming dominance of the distributed computing environment.
4. CORBA also failed to address a specific inter-platform communication protocol.

4.2 DCOM

This is a product of Microsoft Solution for distributed computing. DCOM is object-oriented middleware technology that allows clients and servers in a distributed system to communicate with one another. DCOM is COM extended for supporting objects across a network. DCOM allows objects to be freely distributed over several machines and allows a client to instantiate objects on remote machines. It works similar to CORBA.

WEAKNESSES OF DCOM

1. DCOM is Microsoft-Centric; it often requires some version of Windows.
2. It is not cross-platform independent.

4.3 RMI

Sun Microsystems, with its Java language standard, provides a technique referred to as Remote Method Invocation (RMI), which allows any Java class (object) to be transformed into a distributed component or Java Bean. The RMI effectively makes Java a distributed computing environment. The weakness of this approach is that, it requires both client and server to be Java application.

5. EMERGING TECHNOLOGY FOR INTEROPERABILITY - XML

The Extensible Markup Language (XML), is a standard structured documents and data on the web. It is a product of the W3C (World Wide Consortium), an international industry group made up of about 470 organizations [9]. XML promises to be the solution to

all the problems and weaknesses associated with existing methods of interoperability (i.e. CORBA, DCOM, RMI).

5.1 USE OF XML

XML is likely to become de facto standard for integrating data and content during the next few years, because it offers many advantages such as (1) a mechanism for flexible data storage and data exchange; (2) a means of bridging different technologies such as object models, communications protocols, and programming languages; (3) replacing HTML as the language of choice for designing Web sites; (4) serves as media-independent content to new delivery channels, the Wireless Markup Language (WML) is an example of an XML-based language that delivers content to devices such as smart cell phones and personal digital assistants [10].

5.2 THE BENEFITS OF XML

The benefits of structuring the content sent over the Internet are obvious; they include the following: (1) Search engines could be focused on desired attributes of information only, filtering out the unneeded stuff; (2) Web sites could present views that are appropriate to the devices used to access them; (3) Information could be manipulated intelligently; and (4) Applications can be loosely coupled to be able to interoperate.

6. PROPOSED ARCHITECTURE OF INTEROPERABILITY

The figure 1 below depicts the architecture of database-to-database interoperability using XML while figure 2 shows the generic architecture of interoperability of information systems.

6.1 Architecture for Database-to-Database Interoperability

As shown in the figure 1, a Java Object Translator extracts (by SQL statements – reads) from MYSQL database necessary data and translates the data into XML. As shown, XML data is the input to a second Java Object Translator (via the dotted line on the same LAN), which translates the XML to inject into the Access database (by SQL statements – writes). The functionality is SQL to XML extract, XML to SQL inject using Java Objects Translators.

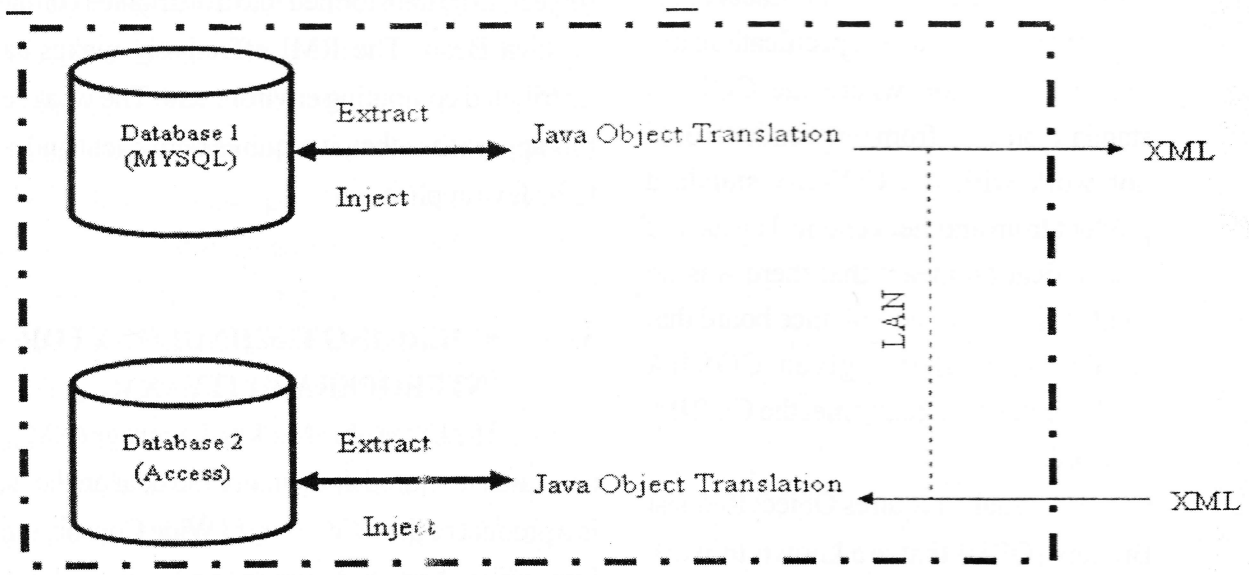

Figure 1: Architecture of Database-to-Database Interoperability

6.2 Proposed Comprehensive Architecture of Interoperability

The figure 2 below depicts system architecture and highlights the integration and interoperability of various information systems. Information System (IS) is an application system with which users interact to perform certain task that dynamically produces output content for display. The master wrapper coordinates schema mapping and message passing among different IS domains, thus aiding IS-to-IS integration. IS wrapper translates and routes messages between its IS and the server. It also provides its IS mapping rules to appropriate links. The wrapper maps commands sent by the system interface to actual system commands and invokes the IS commands. The process involves connecting to the database and creating Document Type Definition (DTD) for the data extracted from the database and saving it as XML format. The Extensible Stylesheet Language Transformation (XSLT) is the part of XSL that turns one XML document into another format that can be recognized by a browser. XSL is used to send XML data to various devices, including handheld, print, or voice output. In this case the XML file is send to the system interface.

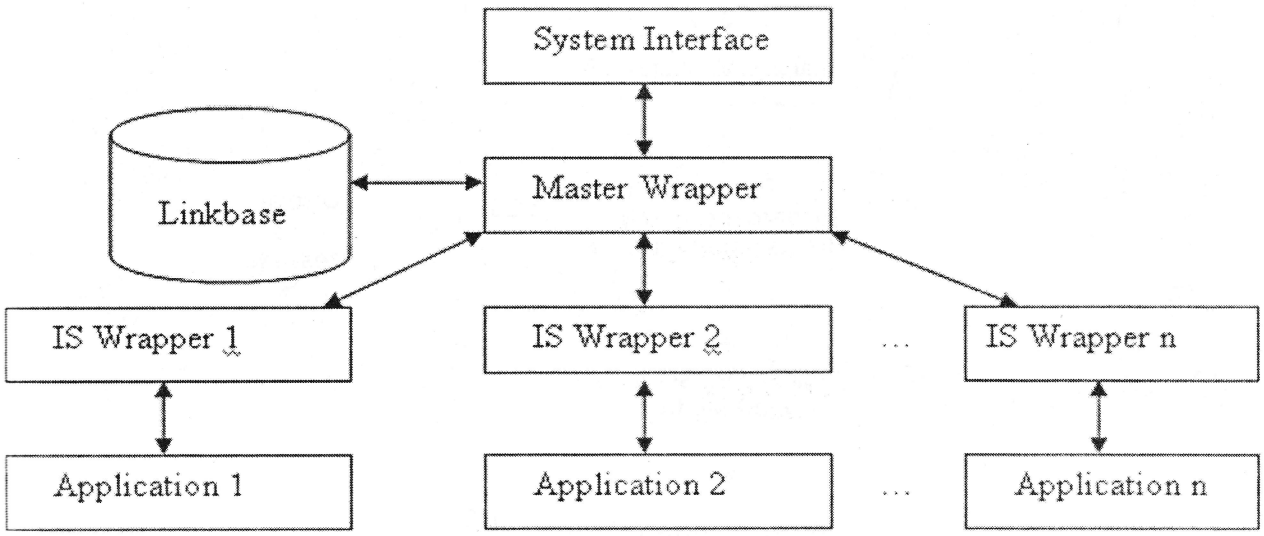

Figure 2: Generic Architecture of Interoperability of Information Systems

7. IMPLEMENTATION (DATABASE INTEROPERABILITY)

Java programming language, eXtensible Markup Language (XML), MYSQL and MS-Access Database were used to implement the prototype software developed. A Java program was developed to read XML document file. The XML file consists of records of a database which can be created by exporting a table in a Database Management System (DBMS) such as MS-Access, MYSQL, Oracle as XML document. A Document Type Definition (DTD) file was also created to define the fields in the database. A Java program was written to read the XML file into a readable form on the screen and equally writing to another database. The algorithm is shown in section 7.1.

7.1 ALGORITHM TO TRANSFORM XML TO DATABASE WITH JAVA

Begin

Invoke Java Packages

(javax.xml.parsers.*, org.xml.sax.*, org.xml.sax.helpers.*, java.io.*)

Initialise variables

Title, Author, Publisher, PublicationDate,

```
Isbn, Edition, Review String;
        If (FileFormat is not XML ) {
          Display Error Message() }
        Call SAXParser function to read XML file
        While not EOF() {
          Assign XML values to declared vari-
ables
          Display record on the screen
          Write record into database
        Loop}
End
```

7.2 FLOWCHART OF XML TO DATABASE CONVERSION WITH JAVA

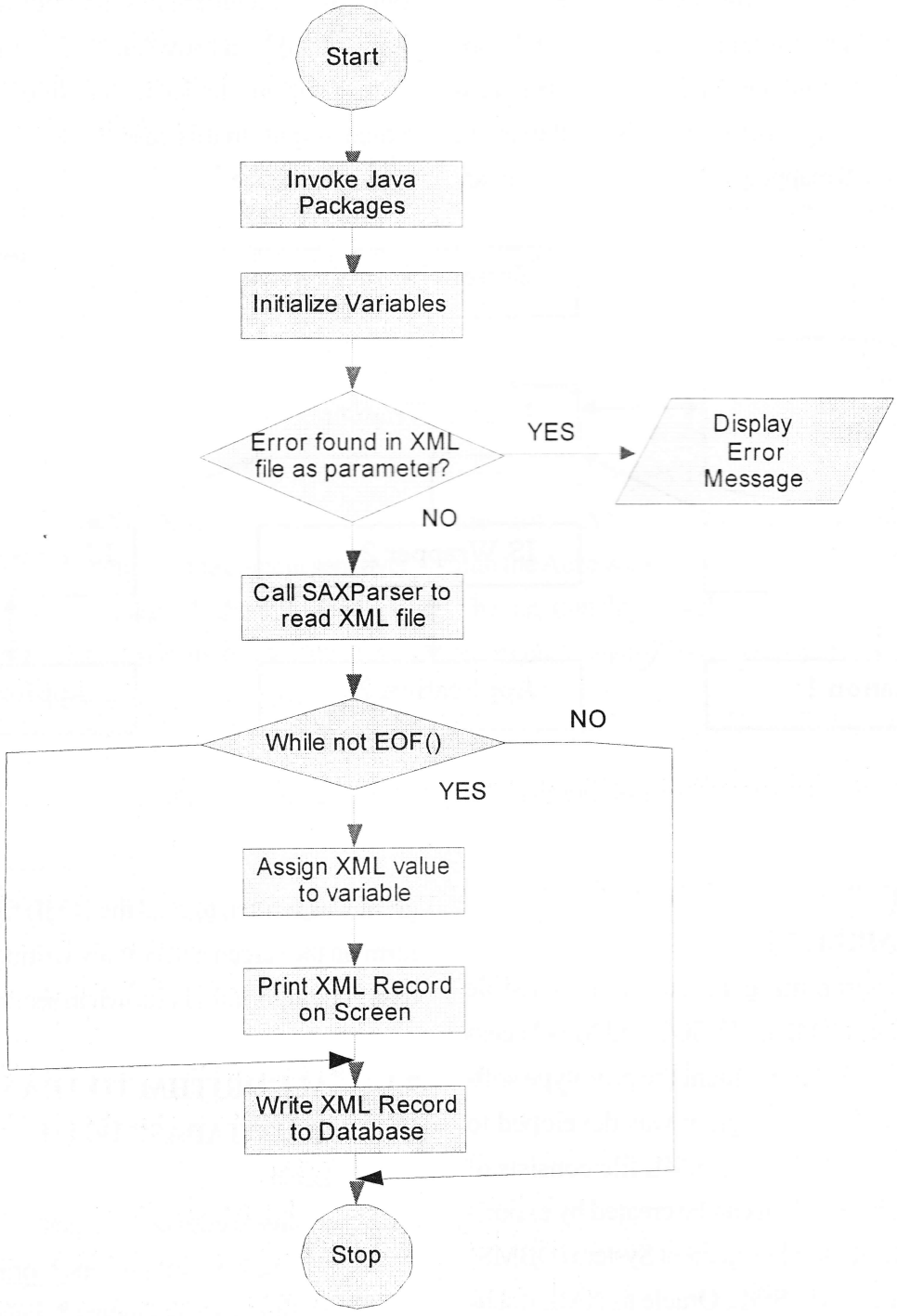

Figure 3: Flowchart of XML to Database Conversion

7.3 OUTPUT DISPLAY ON SCREEN AND IN DATABASE

Figure 4 shows the output display on screen after transformation of XML document file to readable format with Java programming language. Also figure 5 depicts content of the XML document file that has been transferred to another database with Java.

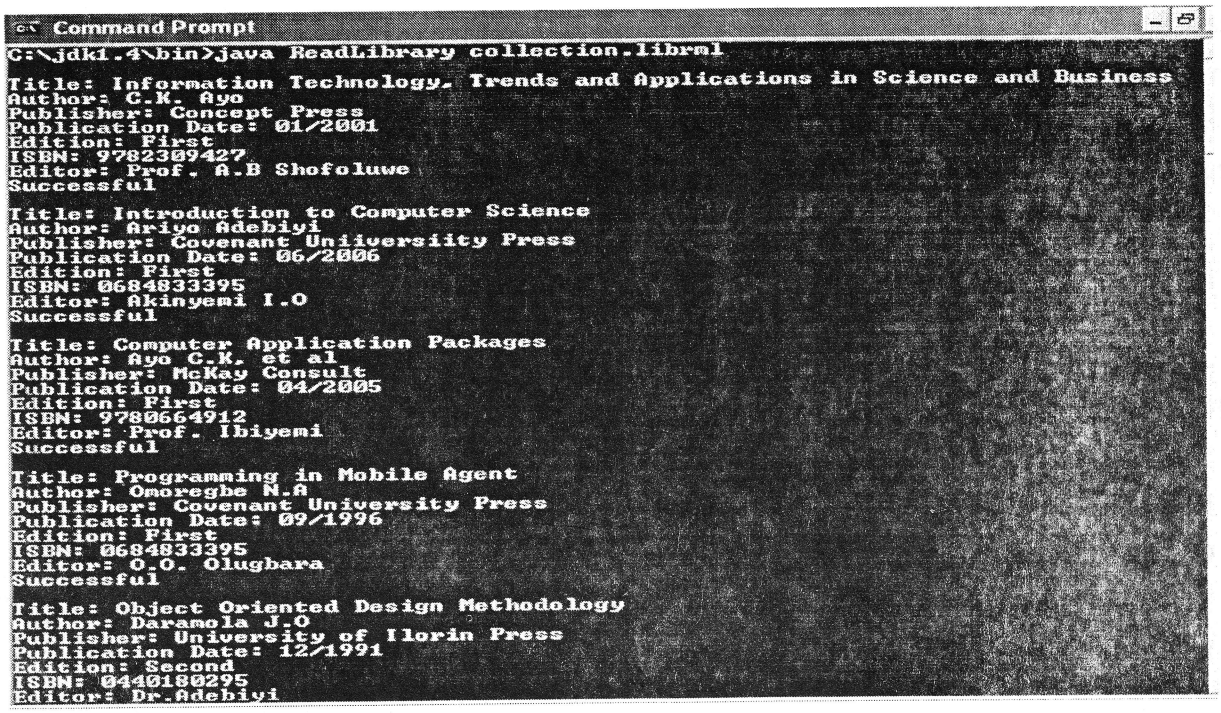

Figure 4: Output of XML file transformation to readable form on the screen

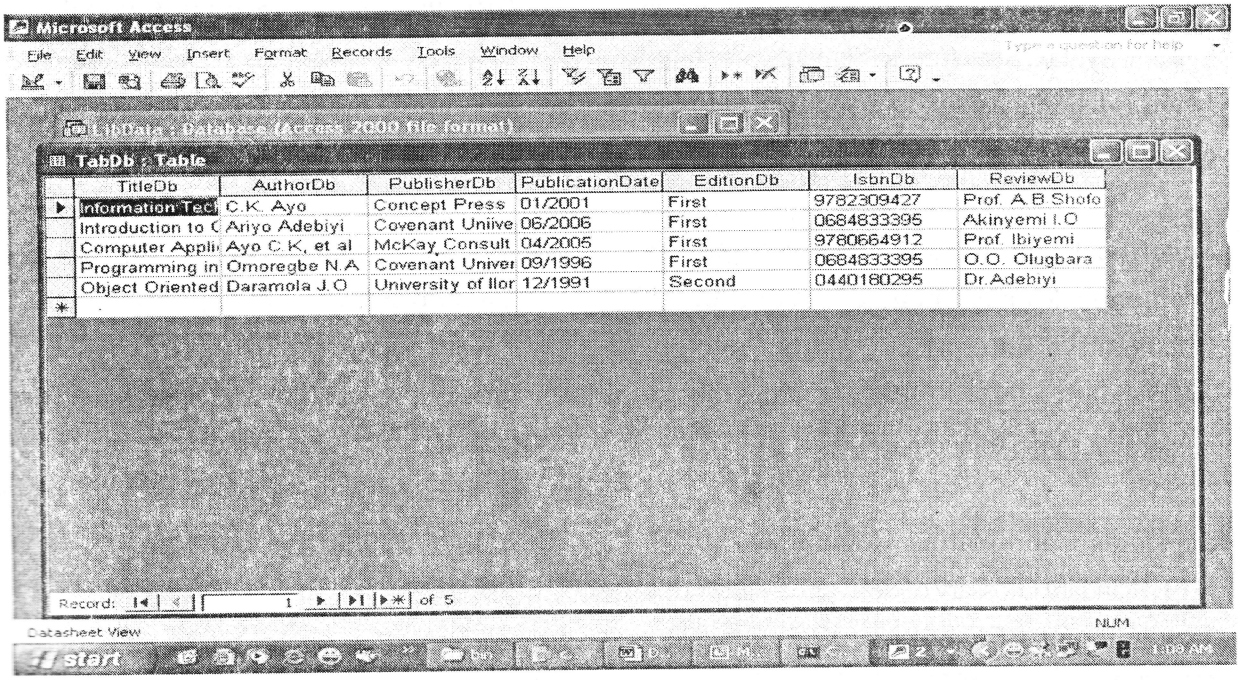

Figure 5: Output of XML file content written to database

8. CONCLUSION

We have presented the concept of interoperability of information systems, also showing the architectural design for proposed solution on interoperability of information systems and development of prototype software for interoperability of information systems and databases using XML. Since, the relevance of integration and interoperability of information systems remain visible on daily basis as e-business is growing rapidly and organizations are thriving to survive in dynamic and volatile business environment. The interoperability of information system will no doubt boost e-business, e-government, e-health, e-learning and so on.

REFERENCES

[1]. Bhavani, T. and Daniel, L.S. (2002)., Using CORBA to Integrate Database Systems in: Judith M.M.(ed.) Enterprise Systems Integration, CRC Press LLC U.S.A 2nd Edition. pp 261-270.

[2]. Amit, P.S. (1998), "O Changing Focus on Interoperability In Information System: From System, Syntax, Structure, to Semantics", <http://www.isdis-cs.uga.edu/ library /download/S98-changing.pdf>

[3]. Severine Blanc (2005), "Interoperability Problems: Management of Evolution of Colaborative Enterprises". <http://interop-esa05.unige.ch/INTEROP/proceedings /doctoral/perpaper/vii-1-blanc.pdf>.

[4]. Standards Coordinating Committee (1990). IEEE Standard Glossary of Software Engineering Terminilogy (IEEE Std 610.12-1990). New York, NY: Institute of Electrical and Electronics Engineers.

[5]. Heileman, G.L. and Jamkhedka, P.A.(2005), "DRM Interoperability Analysis from the Perspective of Layered Framework", <http://www.titr.uow.edu.au./DRM2005/ presentations/drm05-heilman.pdf>

[6]. Adebyi, A.A. and Ayo, C.K. (2006), "Conceptual Framework for Interoperability of Information Systems". Proceedings of International Conference on New Trends in the Mathematical & Computer Sciences with Applications to Real World Problems pp. 119 -130, Covenant University, Ota.

[7]. Ellen, P. and Ellen M. (2004), Programming The Web Using XML, McGraw-Hill Companies, Inc., New York.

[8]. David R (2002), "Distributed Computing: Objects and Components in Enterprise", in: Sanjiv P. (ed.), Architectures for E-Business Systems: Building the Foundation for Tomorrow Success, CRC Press LLC U.S.A pp. 389-400.

[9]. Charles, D. (2002), "XML: A Common Data Exchange Language for Businesses", in: Sanjiv P. (ed.), Architectures for E-Business Systems: Building the Foundation for Tomorrow Success, CRC Press LLC U.S.A, pp. 459-468.

[10]. John van den, H.(2002), "XML – Rosetta Stone for Data", in: Sanjiv P. (ed.), Architectures for E-Business Systems: Building the Foundation for Tomorrow Success, CRC Press LLC U.S.A, pp. 453-458