CHAPTER 1 INTRODUCTION

As XML technologies and their adoption are accelerating across various industries, many software developers and software vendors have a growing need for tools to easily access, manipulate and move data between XML documents and existing legacy relational databases or Relational Databases Management Systems (RDBMS). Over the past few years, XML has become the undisputable standard both for data exchange and content management in IT business world, especially in B2B and B2C applications. However, most of these companies still rely on using relational databases to store and manage structured data for their daily critical business transactions. In fact, major relational database vendors, which also happen to be the giant software companies like Microsoft (SQL Server), IBM (DB2) and Oracle, have ventured and taken great initiatives in researching and searching for single solution to integrate semi-structured XML data with structured data in relational databases. As a consequence, many software developers and information system analysts are now facing the development challenge of converting XML data to relational database format.

This project is aimed at providing an XML-based human-machine readable interface that allows a user to use XML for dealing with semi-structured data for creating, accessing or updating to an existing database of any type that stores structured data. In fact, it is a XML-based middleware interface that sits between XML documents and relational databases for transferring XML data from XML documents to heterogeneous relational databases and vice versa. The proposed middleware interface design should be flexible, robust and easy-to-use innovative technology where it will assist software developers to extract XML documents, and hence allow creating, accessing or updating these XML data to any type of relational databases. Similarly, it is used to marshal and un-marshal data between semi-structured data and structured data using object-relational mapping technologies. Moreover, with this system, any XML document created will be semantically checked and validated for its validity and well-formed format before a consistent XML document is generated for the developers. Furthermore, the interface will execute any required SQL query to create, access or update any specific relational database.

The implementation scope of this project is to study and analyse the possibilities and advantages of the enabling XML technologies in order to implement a working prototype of XML-based middleware interface for the XML users to work with any heterogeneous relational database environment. The proposed project deliverables will not be replacing any of the existing XML middleware or XML-enabled databases but will propose a unified and generic solution of XML technologies without having the software developers and information system analysts to acquire in-depth knowledge of the underlying XML technologies used, such as XML syntaxes, XML schemas, XPath, XQuery or XSLT, etc.

1.1 Problem Statement

Generally, in today's growing XML demands, many people have mistakenly understood that XML is some kind of database format for storing and retrieving data. They do not realise that XML is becoming a widespread lingua franca of data interchange and content management. Looking at the rapid pace of XML technologies and their integration with relational databases such as DB2, SQL Server, Oracle, Sybase, PostgreSQL, etc to support XML features, many software vendors and third parties have also developed various XML tools and solutions for integration of XML data with existing relational databases. In parallel, there are many XML tools and solutions known as XML middleware developed to solve the problem of effective and automatic conversion of XML data into and out of relational databases when integrating XML documents with relational databases. These XML middleware are software that is invoked from an application to perform XML transformation and conversion by importing XML data into relational tables and exporting XML data from databases, they are mostly designed to transfer data between XML documents and relational tables via database drivers such as ODBC, JDBC, or OLE DB. Some have embedded XML validation engine to validate and generate these XML mapping files during the transferring process.

At the same time, major database vendors, such as Microsoft, IBM, Oracle and Sybase, have developed their own tools to be XML-enabled to assist software developers in converting and interfacing XML documents into relational tables. XML-enabled databases contain extensions for transferring data between XML documents and their own data structures. They are used by data-centric applications, except when the database also supports native XML storage' (Bourret, 2000). Generally, data-centric documents as defined by Ronald Bourret are documents which originated both in and outside of the database that use XML merely as a data transport. However, most of them are proprietary under the hood, thus specific to its own database product and implemented using different approach. Moreover, they do not provide a common and unified interface to allow software developers to connect and manipulate heterogeneous relational databases when it comes to integration of XML data with multiple database environments in enterprise application integration (EAI) lifecycle.

Another obvious problem that needs to take into consideration is that these existing solutions do not provide simultaneous connections to several different databases at the same time. Most of them cater only to one connection at one time to one database server and they are only allowed to manipulate the tables that reside in that database server. Another limitation of these solutions is that most of them do not provide a standard XML approach of querying, updating and searching XML data natively from multiple XML documents before transferring to back-end databases. However, some of them are already aware of this potential demand and are working towards this direction by proposing to implement some proprietary XML query language such as XQL, Quilt, XML-QL and so on. This is true especially in native XML databases which have already implemented these XML query languages to store and retrieve XML data natively. However, in October 1999, W3C established XML Query Working Group to propose a standard query for XML data known as XQuery language; it is still under W3C working group drafts and suggestions.

Unlike most middleware, only few middleware support inserting, updating or deleting data from a single XML document to relational tables. Moreover, most of these middleware are either provided in Java classes, off-the-shelf components or APIs in programmatic approach instead of providing a graphical user interface where any level of user can use the system. Usually, the data transformation from XML to SQL is done automatically once the user has selected the XML document to transfer, and users are only allowed to define or change any data during design time but not execution time. Hence, they should provide parameterised approach because this flexibility enables user to make any changes during execution time without having to change any application coding.

As there is increasing number of software developers shifting to XML-based and relational database development paradigm due to business requirements, subsequently this scenario will create common development issues when integrating XML and relational database technologies. Developers are facing these challenges because they do not have a consistent tool but have to rely on own individual coding efforts to code for this infrastructure to enable them to manipulate multiple XML documents using XML query languages such as XQuery before transferring to heterogeneous relational databases. Therefore, there is no consistency in their applications. Even though it is impossible to design a system that model all required features for a given set of XML documents, but it is usually worthwhile to represent similar kinds of XML documents using similar approaches and design the overall framework using reusable components. Consistency helps software developers in implementation of XML documents, simplify maintenance, and even aid in automating XML document modeling and mapping processes. To achieve this objective, the proposed prototype is aimed at solving these arisen problems by providing a solution to simplify the interoperability between XML data and relational databases as such to boost the productivity of the software development teams.

1.2 Research Objectives

As mentioned earlier, XML has become a widespread standard over the past several years for both data exchange and content management. It is known as a meta-language for both human and computer readable and it can also be understood easily by any software application. Having delved into the problems stated in the previous section, there is a growing need to design such interface which makes possible alternatives and solutions to the stated shortcomings to be unearthed to overcome the weaknesses mentioned.

As a result, the goal of this project is to study the needs and methodologies of implementing a middleware prototype of an XML-based interface that will integrate with existing heterogeneous relational databases in order to solve the current limitations as described above and also to enhance the development processes of software developers who are working in this area. Below are the main objectives of this project:

- To study the need for such middleware as a common interface to integrating XML with existing heterogeneous relational database environments.
- To study XML technologies such as XQuery and XML mapping to relational database structure, and vice versa.
- To specify the system requirements and specifications, system analysis and design using object-oriented system development and RUP methodologies.
- To develop a working prototype using 100% Java and object-oriented architecture.
- To provide connecting, querying and manipulating of XML data between XML documents and heterogeneous relational databases.
- To provide bi-directional data transfer between XML documents and relational tables.
- To provide graphical XML view over relational database and XML documents.
- To provide simple manipulation of XQuery expressions via XQuery wizard.
- To provide XML validation engine to semantically check for correctness of generated XML documents.

- To evaluate the developed prototype and refinement of the prototype system.

Consequently, the research result of this project is the deliverable of the XML-based middleware interface designed to integrate XML data with heterogeneous relational databases. This deliverable will be tested with several relational databases such as Microsoft SQL 2000, MySQL and Oracle. Its flexibility, user-friendly and intuitive capability is designed to connect, create and manipulate XML data with existing relational databases, automatically generate semantically checked XML documents and their DTDs. Thus, developers or information system analysts can query, create, manipulate and retrieve XML data regardless of any database, or vice versa, in just few clicks without going through the programming learning curve and platform compatibility issues.

1.3 Scope of Research

The scope of this project is to design and implement a flexible and user-friendly graphical XML-based middleware interface that allows users such as software developers or information system analysts to connect, create, query and manipulate XML data on the fly between multiple XML documents and heterogeneous relational database environments. Moreover, this project takes into consideration of providing simple XML-based interface, which is both human and computer readable, that will allow its users to use XML in dealing with semi-structured data for creating and manipulating these XML data to integrate to any existing relational database. Besides that, this system also provides a graphical interface that is able to transform relational tables and fields into XML view following several stages of mapping from XML documents to relational databases between XML elements or attributes of XML schemas with relational database structures such as tables and records. These XML mapping technologies provide software developers a comprehensive access and manipulation to heterogeneous relational databases without the need to learn complex XML technologies and SQL programming.

The proposed system's architecture is based on object-oriented design and Java architecture, thus it will be flexible and portable across WINDOWS and UNIX platforms. It provides transparent mapping of XML elements and attributes to underlying relational database structures and data binding object interfaces that enable software developers to create and serialise objects which will represent the records in the database. Once the connection to a relational database is established and the mapping is in place, the users will have a full bi-directional access of XML data between XML documents and relational database systems. Here, the proposed system will manipulate XML elements and their nested elements via XML APIs such as Document Object Model (DOM) or Simple API for XML (SAX), Java XML API, or simply querying and updating database records via this XML interface. Any XML document generated through this interface will be semantically checked for its correctness and well-formed format before a valid and well-formed XML documents are generated. Besides that, respective SQL queries can be generated for creating new tables, accessing and updating tables or records.

The proposed system will offer a rich yet flexible XML interfaces that enable software developers and information system analysts to manipulate semi-structured XML data and relational databases back and forth without having to learn any in-depth specific programming skill. Furthermore, this system manages its connections to multiple relational databases for better and efficient XML integration with relational database systems. As a result, this proposed system hopes to provide a simplified, ease-of-use and consistent XML middleware interface that is able to integrate XML data with any relational database systems.

1.4 Significance of Research

The significance of this research is to identify suitable design criteria for developing a XML-based middleware interface integrating with heterogeneous relational databases. In addition, it includes the identification of suitable XML and Java technologies to be used in this XML-based middleware development. From this research conducted, a working prototype will be implemented at the end of the development using the proposed design and XML enabling technologies identified in the following chapters. With the availability of this XML-based middleware interface, software developers and information system analysts can easily manipulate XML data between multiple XML documents and heterogeneous relational databases, either to create, query or update XML data to relational tables without having to acquire any in-depth XML or SQL programming skills.

Subsequently, it is hoped that this prototype will not be replacing any of the existing XML middleware or XML-enabled databases in the market, but serve as a basis milestone to be used as a generic XML-based middleware interface that implements consistent and generic XML framework approach for integration between XML documents and heterogeneous relational databases. The success of this prototype system is expected to increase the power of integration and development productivity among the software developers and information system analysts who have to work with large-scale of XML documents that need to be integrated to multiple relational databases and vice versa in an integrated application enterprise world.

1.5 Dissertation Outline

This dissertation outline is divided into six separate chapters. Firstly, Chapter 2 will be concentrated on the literature review of this dissertation. The review will cover on the relationship between XML documents and relational databases, and significant benefits of XML technologies in software application development lifecycle. In addition, brief reviews are done on the enabling technologies such as XML mapping definition and Java technologies that are used to transfer XML data in and out of XML documents and relational databases.

Chapter 3 describes the overall methodology used to conduct this dissertation. It discusses all the essential steps taken in the software development life cycle (SDLC) for this system. These steps include conclusions from the literature review, system design and analysis, implementation, testing and evaluation, and future enhancement work. Every design and strategy that is used to implement this system will be documented here conceptually.

Chapter 4 discusses about the analysis and design of the proposed system. It analyses the techniques used in more technical details and the actual design architecture are clearly defined in this chapter. Apart from that, it explains how the design of the system is translated into a working model using an appropriate programming language such as Java language and a modeling language such as UML class diagrams.

Chapter 5 thoroughly covers the implementation, testing and evaluation of the final prototype. In this chapter, it describes the implementation of the system using UML sequence diagrams and component diagrams. It also includes the system analysis of the test results in terms of performance, efficiency, flexibility and functionality of the system. Moreover, evaluation is performed to determine whether or not the final prototype deviates from the system specification and design, and also whether the objectives of the dissertation have been achieved. In addition, the test results from using various relational databases such as Microsoft SQL, Oracle, and MySQL to integrate XML data with XML documents are also observed in this chapter.

Last but not least, the conclusion of the whole dissertation research is addressed in Chapter 6, together with possible future enhancement works, strengths and limitations. Here, appropriate strategies and techniques, which might be possible to enhance the future development works, are also discussed.