Volume 4, No. 4, March-April 2013
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Enhanced Database Migration Technique Using XML

V.Hemalatha*, Alok Ranjan
Assistant Professor, CSE
Canara Engineering College Mangalore, Karnataka
hemaprem2983@gmail.com*, mibalok07@gmail.com

Abstract: XML becomes a de facto standard for representing and exchanging data over the Web. It is designed to structure and carry data in a sensible way, thus helping programmers and web developers manipulate the data easily and efficiently. With the tremendous growth of XML data on the Internet, an efficient database system becomes necessary to maintain it. There are many Internet applications that produce and consume large volume of semi-structured data for the Web applications. It will be viewed as a large distributed semi-structured database with XML as its model, and the issues related to storing, managing, and querying XML data become more crucial.

Keywords: semistructured database, XML database migration, XML code generation, off-host processing, physical threats, logical threats, Resource Description Framework (RDF)

I. INTRODUCTION

Traditional databases that force all data to conform to an explicitly specified rigid schema are inefficient to maintain irregular XML data, thus, new data models, called native XML database management systems, are introduced. Currently proposed native database systems mainly focus on designing database architectures and its components, but they have not addressed how an efficient user interface can be designed to support users to interact with their databases. Figure 1 represents the XML layers.

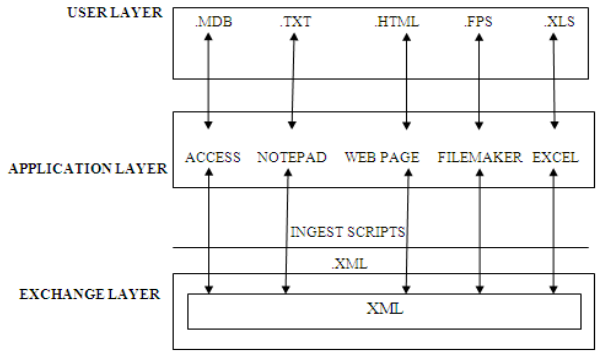

Figure 1. XML layers

Threats on XML database security can be grouped into two different categories, physical and logical. Examples for physical threats are forced disclosure of passwords, destruction of storage devices, power failures, theft and logical threats are denial of service, disclosure of information, and modification of data. The most common way to prevent this physical threat is limit the access to the storage devices and put backup and recovery procedures in place [1]. Logical threats are unauthorized logical access to information. This is usually through software.

II. XML DATABASE MIGRATION

It is a process of migrating the existing databases into XML format. Conversion of databases to XML file format, the databases like Ms-Access, Oracle and MS-SQL connects in the network or the oracle in the personal computer can be connected for the generation of XML code. Let us consider, user runs a firm with three companies with different database's respectively. User planned to mingle the entire database's of the firm in to a single database.

The user wants to migrate the entire database in to a single database. Migration of a database to another database is a huge process and a complex one. In such case, the entire database is converted into a XML file and then to a single database. By doing so, the conversion is been made easier and the native database also will remain unchanged. Only the authorized person is allowed to access the databases. The user is been prompted for username and password and then the user can be allowed to select the type of database.

After selection, the user is been prompted for database name and then corresponding tables present in that database is been displayed from which the user can select the table which the user wants to convert. Before conversion, the user can view the table from that the user can check whether the correct table is been selected. The table is been converted and a message will be displayed to the user about the conversion.

The converted file will be in the XML format and it is been previewed to the user in the database format after conversion. Here, the user can check whether all the fields in the table are converted. While converting the data's in the tables, the tables present in the database remains unchanged and the fields from the tables retrieved are subjected for XML conversion.

XML Migration is project to convert the existing Database like Oracle and MS-Access into the XML database. XML is a text formatted flat file, so it has an ability to store any kind of data from the application system. On using the XML as a backend, it reduces the memory occupied in Internet and there is no need to pay amount separately for XML database as it is file and consumes less memory [2,3].

III. NEED FOR DATA MIGRATION

Whether for permanent change to a new environment, or for the operational convenience of "off-host" processing, the ability to move data between computing platforms of

different types increases the flexibility of enterprise information technology operations significantly. But the complexity of moving data between unlike platforms has been a barrier to exploiting data assets in this way.

Different platforms have incompatible volume metadata formats, file system metadata formats and data formats in application files. To transfer data between platforms, IT departments have had to choose between network copies and copying data to tape on the source platform and restoring tapes on the target platform.

Both of these options consume significant time and resources. As a result, many IT departments continue to run applications on less-than-optimal platforms because migration to more suitable environments is believed to be too resource-intensive[4]. Others forego the business benefits of off-host backup, data mining, and testing with live data because they believe that copying large data sets would result in unacceptable application downtime [4]. In effect, each data set becomes captive to the server platform that processes it.

Before migration, the following analyses must be performed:

a. Analyze and define source structure (structure of data in the legacy system)
b. Analyze and define target structure (structure of data in the new system)
c. Perform field mapping (mapping between the source and target structure with data cleansing)
d. Define the migration process (automated vs. manual)

IV. DATA EXCHANGE PROBLEM IN XML

Data exchange is the problem of migrating the contents of a data source to an instance of a target schema that reflects the source data as accurately as possible. The data exchange problem has become very important with the creation of XML, a data format that was created for the very purpose of data exchange.

The basic problem in the data exchange problem is to recognize which data from the source goes where into the target. In this sense, it contains partly the schema integration problem, but with two major differences. First, the target schema is already present, there is no need for its creation. Second, contrary to what is the case with the schema integration problem, the target schema has its own set of constraints that must be dealt with when performing the data migration.

A. Complexities and Challenges:

Various types of databases are being used in Real-time. The complexity arises when a there is a need for inter-operation between databases. At this stage all or part of databases must be migrated to a common type of database.

If the databases are too large then surely it will be an overhead. These databases occupy some part of memory in the server. The site owner should be paid some amount of money to maintain that database. The amount should be vary depending upon the database size and also have a lot of problem to maintain the database. Replication of database and backup of database has to perform on a time to time basis as it consumes large amount of memory [5,6].

There are two reasons for data exchange problem. First, relational databases have very little structural data in contrast with XML files, that in many cases the structure is more than the actual data. This means that there is more structural information which can be exploited to find associations between data items. Second, with the use of namespaces, RDF and ontologies, data migration in the future will be fast, effortless and error-free [7].

V. MIGRATION TECHNIQUES

The database migration technique improves the performance of applications. There are following steps for database migration technique [8,9].

a. Create a comma-separated values (CSV) formatted or XML Spreadsheet 2003 or higher version source files that contain the source data.
b. Create the data maps, or use the existing data maps. This step is not required if you use auto-mapping.
c. Create the import files and read the content of the CSV source files into the associated import files.
d. Parse the import files.
e. Transform the parsed data.
f. Upload transformed data into the target Microsoft Dynamics CRM server.

VI. PROPOSED METHOD

In the proposed method, there are two assumptions. First, every element in an XML file is semantically related to its parent and children, and thus by transitivity also to its ancestors and its descendants. Second, any two elements in an XML file are in fact semantically related, even if one is not in the ancestors-descendants path of the other one, since they will both share a common ancestor.

The input to the algorithm is the source XML file, its XML Datasource Schema, the target XML Datasource Schema, and a file specifying the mappings between element names in the source and the target Datasource Schemas. In order to migrate the data, the algorithm must in each step have knowledge of the elements it will import from the source file in order to insert them in the target file. To do that, the algorithm traverses the target Schema and for every node encountered, fetches the appropriate node(s) from the source by creating the appropriate XPath expression.

The enhanced data migration process consists of following steps,

a. Server Accessing Module
b. Retrieval of tables
c. XML Code generation
d. Preview of XML file

A. Server Accessing Module:

Oracle and SQL server has its own username and password, for accessing the database file. Both have high security. Without correct user name and password, it can't access oracle and SQL server database file [10]. In converting databases to XML file format, the database like MS-Access, Oracle connects in the network or in the personal computer for the generation of XML code.

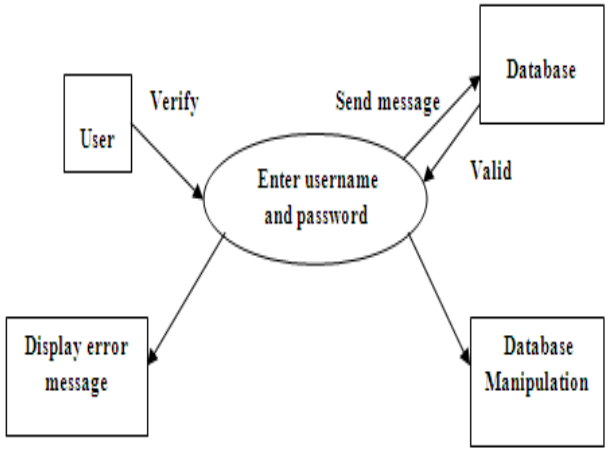

Figure 2. Server accessing module

It should provide authentication for user and protect the server database from unauthorized access. MS-Access database can be connected from the local drivers, floppy or compact disk. The function of server accessing module is shown in Figure 2.

B. Retrieval of tables:

By selecting the each table in the database, the corresponding structure of the table is seen. In particular fieldname, data type, size, precision and scale are displayed. They are collected for verifying the cascading style sheet. If there is any need for modifying the database, then by selecting the table, manipulation like insertion, updation and deletion operations can be done.

This helps to make a necessary manipulation and important changes at that time of migration. After authentication, the user is allowed to select the database type and prompted for database name[11]. The User is made to select the table for which the conversion is to be carried out. It can also be previewed for the case of reducing mal-selections. The process of table retrieval is shown in Figure 3.

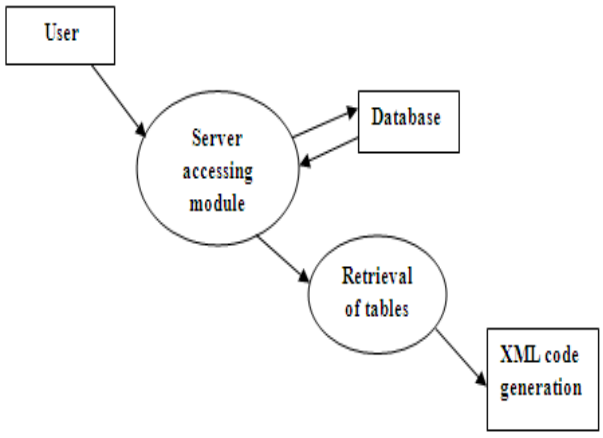

Figure 3. Retrieval of tables

C. XML Code Generation:

The data's in the table are selected and converted into XML file format. The selected table is been implicitly assigned to a temporary file for the conversion process. The assigned file is converted into a XML file by using the appropriate code.

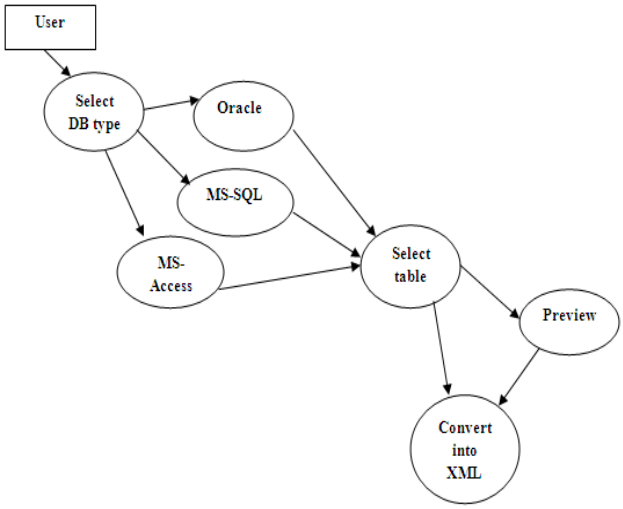

Figure 4. XML Code Generation

After conversion, the original data's in the database will remain unchanged. Individual tags are created for each and every field in the table retrieved from the database, specified by the User[12]. This module is used to generate the XML code for the database and it is shown in Figure 4.

D. Preview of XML file:

The file name of the XML is been prompted to the user so that he can view in XML format. It can check whether all the respective fields of tables requested by the client are converted. If any error occurs, it can modify the generated XML file and the preview of XML file is shown in Figure 5. There will be no change in the source file.

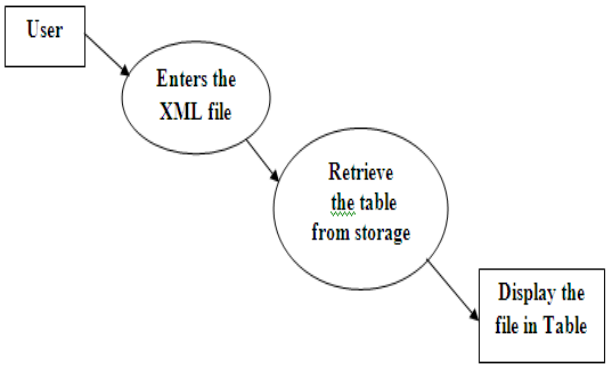

Figure 5. Preview of XML file

VII. CONCLUSION

The XML is a text format flat file and has an ability to store any kind of data in different operating system. XML file format support different platform like Linux, Windows etc. Even though source data's crashes, the database can be backup from the XML file. By migrating into common file format, the database size and accessing time has been reduced with less human resource.

The databases are migrated to XML file. In future this can be enhanced into multiple databases. Here, the semantic constraints present in the source databases are not included in the conversion. It can also be included in the conversion process in near future.

VIII. REFERENCES

[1]. Bo Chen, David D. Linz, Harry H. Cheng, XML-based agent communication, migration and computation in mobile agent systems, The Journal of Systems and Software 81 (2008) 1364–1376

[2]. Lucas Zamboulis, XML Schema Matching & XML Data Migration & Integration: A Step Towards The Semantic Web Vision, Technical Reports, October 21, 2003.

[3]. K. M. Win, W. K. Ng, and E. P. Lim. ENAXS: Efficient Native XML Storage System. In LNCS 2642: Proceeding of the 5th APWeb Conference, X'ian, China, April, 2003, pages 59–70. Springer Verlag.

[4]. H. Meuss, K. U. Schulz, and F. Bry. Visual Querying and Exploration of Large Answers in XML Databases with X2. In Proc. 19th Int. Conf. on Data Engineering, Bangalore, India, Mar. 2003.

[5]. K. M.Win, W. K. Ng, and E. P. Lim. A Survey of XML Data Storage and Indexing Techniques. In Proc. 1st Int. Conf. ICCA, Yangon, Myanmar, Jan. 2003.

[6]. S. Sipani, K. Verma, S. Chandrasekaran, X. Zeng, J. Zhu,D. Che, and K. Wong. Designing an XML Database Engine: API and Performance. In Proc. 40th Annual ACM-SE., Raleigh, USA, Apr. 2002.

[7]. H. V. Jagadish, S. AI-Khalifa, L. V. S., A. Nierman, S. Paparizons, J. Patel, D. Srivastava, and Y. Wu. TIMBER: A Native XML Database. VLDB Journal, 2002.

[8]. L. Popa, Y. Velegrakis, R.J. Miller, M.A. Hernandez, and R. Fagin. Translating Web Data. In Proc. VLDB'02, pages 598–609, 2002.

[9]. Joshi, J., Aref, W., Ghafoor, A., and Spafford, E. Security Models for Web-based Applications Communications of the ACM, Feb. 2001

[10]. K. D. Munroe and Y. Papakonstantiou. BBQ: A Visual Interface for Integrated Browsing and Querying of XML. In Proc. 5th IFIP 2.6 Working Conf. on Visual Database Systems, Fukuoka, Japan, May 2000.

[11]. C. C. Kanne and G. Moerkotte. Efficient Storage of XML Data. In Proc. 16th Int. Conf. on Data Engineering, San Diego, CA, Feb. 2000.

[12]. Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic Integration of Semistructured and Structured Data Sources. SIGMOD Record, 28(1):54–59, 1999.

[13]. R. Goldman and J. Widom. Interactive Query and Search in Semi structured Databases. In Int. Workshop on the Web and Databases, pages 52–62, 1998.

