An Approach to Heterogeneous Database Migration

Aditi V. Parab and Onkar Dhane
Department of Computer Engineering, SIT Lonavala
aditi.parab1@gmail.com
engg.onkar@gmail.com

Anuja Deshmukh and Aakash Mahajani
Department of Computer Engineering, SIT Lonavala
danuja92@yahoo.com
aakash_mahajani@yahoo.co.in

***Abstract* - Main issue in the database migration is the work of data organizing, analyzing, accuracy and integrity. In this paper we have tried to eliminate these problems, by making use of XML file as an intermediate. XML supplies a neutral platform for the information description. As the information is mostly stored in RDBs, we have decided to create a system to export the information from a DB into XML, and to import XML documents into any RDB management system. Till date systems were available to covert data from one DB into an XML file, in our paper we have given more emphasis on the conversion of the XML file into the required target DB. Our system has enabled us to transform data from one DB into another using XML as an intermediate representation. Main aim of our paper is to generate a system to automate the transference of data between databases. The proposed system performs conversion of DBs like Ms-Access, MS-SQL, Oracle, and MySQL.**

Keywords: XML, RDB, SQL

I. INTRODUCTION

With the help of technology, we can collect huge amounts of scientific data. Collection is not a problem, but however the work of organizing, analyzing and combining the data is a very tedious work. To join datasets from various different databases, provided by different vendors is sometimes not possible. To combine such different datasets we need to have a common platform, i.e., we need to export the records or the datasets into a common format. Such a common platform can be achieved with the help of XML [Extensible Mark-up Language] schema. As we all know that XML has been widely accepted and used as the universal format for exchanging data worldwide. The reason behind that is the XMLs capability to support flexible representations of data. XML has various features which makes it more efficient

when compared to other schemas. For example: it is not dependent on any platform; it requires less administrator support; makes use of less memory space; has very low maintenance cost; also considered as self descriptive language. Some of its distinct features are: it is free from the data losses; comparatively more secured; has faster access ability.

In current scenario various types of databases are present. Many a times, it is essential to export data from one type to another type of database. In such situations the process of data migration is required. This process includes following basic steps-

a. **Extract**: It extracts data from the source database.
b. **Transform**: Transforming into the intermediate language.
c. **Load**: Migrating to target database.

For conversion of one type of database to another type, XML is the most appropriate intermediate that must be used. For such conversions, the end user need not be aware of any specific language. An end user with little knowledge of technical language will be able to perform the conversion with the help of our system. User just needs to have basic knowledge of how to operate a computer. The basic goal of our paper is to provide the end user with a system that can migrate from one form of database to another and vice-versa.

II. LITERATURE SURVEY

A. *Work of Carey, D. Florescu, Z. Ives and Y. Lu in 2000*

Existing System- forced to deal with the data source's particular (e.g., object relational) schema and query language.

Proposed System- solution provides a uniform XML-based query interface over an object-relational database that allows users to query and (re)structure the contents of the database as XML data, ignoring the underlying SQL tables and query language.

Remark- restricted for the users who prefer to work in pure XML environment; so many other users won't find it useful for them.

B. *Work of T. Nasser, K. Kianmehr , R. Alhaj and M.Ridleyin in 2007*

Existing System- to generate XML documents from the object-oriented database and XML schema whether it's flat or nested using GenXMLDoc algorithm.

Proposed System- presented transforming mapping method in both direction, OODB into XML and XML into OODB. They used backward engineering to transform from XML schema to object-oriented database.

Remark- gives the user the ability to choose type of XML schema and to trace the result of every phase of the converting process. On the other hand, it is complex.

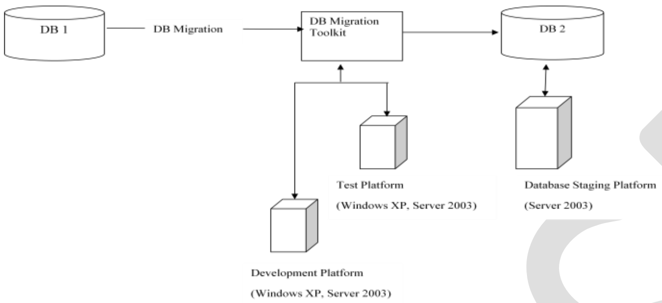
C. *Work of F. Furlan and R. Mello in 2009*

Existing System- ignore the conversion of complex structures, like sets and lists, and none of them provides a treatment for multiple inheritance relationships.

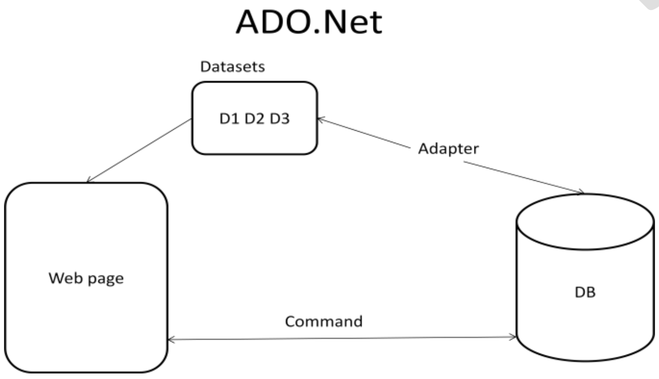
Proposed System- a rule-based approach for converting object oriented (OO) database schemata to XML schemata, as well as an algorithm that defines the application of the rules and deals with multiple inheritances.

Remark- proposal considers the mapping of all OODB model concepts to equivalent data structures in XML.

III. SYSTEM ARCHITECTURE



Conversion of one form of database to another form is taking place. For this we make use of the DB Migration toolkit, which comprises of two platforms- the Development platform and the Testing platform. How the conversion will actually take place is shown in the figure given below-



Two types of connections are possible- Connection oriented and Connection-less. In connection oriented method web page directly sends a request to the database for the required data through the command () function. In this the connection to the database needs to be on at all times. However in connection-less method respective adapters of different databases is used. When certain data is required from the database the adapter fetches it and deposits in the datasets through the fill () function. These datasets are nothing but virtual databases. In this the connection to the database need not be always on as we can get the data from the virtual database.

Front end of our system: ASP. Net [HTML, DHTML, XML, CSS and JAVA SCRIPT]

Back end of our system: C#

Data Providers in C#:

- MS-SQL- system.data.SQLClient
- MS-ACCESS- system.data.oledb
- Oracle- system.data.oracleclient
- MySQL- MySQL.data.MySQLClient

Adapters:

- SQL data adapter
- Oledb data adapter
- MySQL data adapter

IV. SYSTEM FLOW

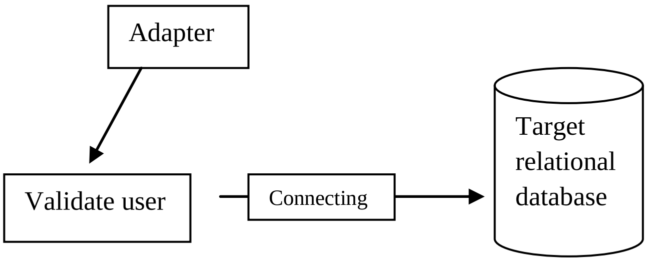
Validation- User needs to register them. After registration user is provided with the username and password. User's login is validated through our database in which username and password are stored in encrypted format.

Select source database- Now user is allowed to access the database and asked to select the source database for migration. Source database consist of MY SQL, MS Access, Oracle, SQL server.

Choose appropriate set of XML documents- XML documents are generated from the source database. It is necessary for the user to choose an appropriate set of XML documents. We also need to identify the links and dependencies between XML documents. There are two types of entities: Superior and Subordinate. Superior entities are not dependent on other entities and will be migrated in the first phase of migration. Subordinate entities will represent database table, depended on main tables.

Loading of XML document- For migration to target relational database we need to load all selected XML document. Documents get loaded according to entities. Superior entity gets loaded first and forms main database tables. The names of subordinate entity will consist of the name of superior entity and subordinate entity. Subordinate entity forms dependent tables with appropriate foreign key.

Connecting to target database- With the help of respective adapter connection with target relational database is formed.



Analysis of XML document and creating relational schema of target database- This step analyze XML documents and based on that automatically create relational schema of the target relational database. Relational schema gets generated based on superior and subordinate entities.

Inserting data into target relational database- This is the last step of migration, XML documents are inserted to the target database. It consists of two steps:

1] Inserting data to main database tables.

2] Inserting data to dependent database tables.

V. PROPOSED SYSTEM

In our system we are going to emphasis on the conversion of an XML file into the required target database. Till date we have systems which convert data of one database into an XML file. With the help of our system user can convert the source database into four most used databases [i.e. MySQL, Oracle, MS-Access, and SQL Server 2005] in one pass only. It is also possible to convert only selected tables if the whole database conversion isn’t required. As the system can be logged on only with the help of a valid username and password, system will be secure. It would allow only the authenticated people to access the converted file. Before conversion, user can view the table, thereby eliminating unnecessary conversions. Even after the conversion of the data present in the table of the source database, the original tables are retained. Hence, no additional backup facility is required. To eliminate the problem of type of data to be used, XML flat file is used. It supports all types of texts, characters and symbols. To eliminate the problem of XML file size, we would be making a directory and keeping all the respective XML files in it. Hence, when the conversion of entire source database is required into the target database we just need to pass the address of the directory. The semantics of the relational database are captured with the help of EER Model [Extended Entity Relationship Model] and then mapped to the XML schema.

VI. ADVANTAGES

This approach uses XML as the intermediate language and mapping will be done in the following manner:

Source Database --> XML --> Target Database

- Since there is no direct database conversion and xml is used as the intermediate medium, there is a huge reduction in complexity and in the amount of time required.
- With the help of view functionality, authorized users will be able to view the tables before their conversion.
- No restriction to XML text formatted Data.
- The need of scripting languages is eliminated.
- Security is guaranteed.

VII. FUTURE SCOPE

Semantic constraints present in the source DB can be included. It can be enhanced to multiple databases. Reusability is possible. XML converted documents are easy to transfer over the internet. One can try to include data like images, audio and video.

CONCLUSION

In this approach we have discussed and addressed and implemented the possibility of heterogeneous database migration. We have used XML as the medium for mapping of databases thus overcoming the tedious and time consuming task of direct database to database conversion. We have achieved the goal of one to many database mappings. In the future another language similar to XML can be developed but with the added functionality of carrying audio, special characters and video which XML cannot carry.

ACKNOWLEDGEMENT

We express our hearty gratitude towards our guide Prof. R.S.Shishupal, Head of Computer Department Prof. T.J.Parvat and Principal Dr. M.S.Gaikwad of S.I.T. Lonavala for guiding us to understand the work conceptually and also for their constant encouragement.

REFERENCES

[1] M. Carey, D. Florescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita and S. Subramanian, “XPERANTO: Publishing Object-Relational Data as XML, Proc. of the International Workshop on Web and Databases, 2000.

[2] M. Carey, J. Kiernan , J. Shanmugasundaram, E. Shekita, and S.Subramanian, " XPERANTO: A Middleware for Publishing Object-Relational Data as XML Documents," Proc. of the 26th International Conference on Very Large Databases, 2000, Pages [646 -648].

[3] T. Naser, K. Kianmehr, R. Alhaj j and M. J. Ridley. “Transforming Object-Oriented Databases into XML”, IEEE 1-4233-1500-4/07, 2007,p[600- 605].

[4] T. Naser, R. Alhajj and M. J. Ridley. “Flexible Approach for Representing Object Oriented Databases in XML Format”. In: X International Conference on Information Integration and Web-based Applications Services (iiWAS), Linz, Austria, Nov 24-26, ACM, 2008. [p 430-434].

[5] T. Naser, R. Alhajj and M. J. Ridley. “Two-Way Mapping between Object-Oriented Databases and XML”. Informatica 33 2009 [p 297 -308]

[6] Sacks-Davis, R., Arnold-Moore, T., Zobel, J., Database Systems for Structured Documents. IEICE Transactions on Information and Systems, Vol. E78-D