www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 10 October, 2014 Page No. 8647-8650

Hetertogenous Database Migration using ODTDM Supported with SAX and SDM algorithms

Priyanka Talole[1], Mayur Talole[2]

[1] Bachelor of Engineering,
Department Of Computer Engineering,
Nutan Maharashtra Institute Of Engineering And Technology,
University Of Pune, India
tpriyanka123@gmail.com

[2]Department of Electronics Engineering,
Vishwakarma Institute of Technology,
University of Pune, India
mayur.talole@outlook.com

Abstract: This paper proposes to describe an approach used for migration of historical database from one database to required database, and setting up a daily feed from the product into the system data to have previous day reports available for current day trading. Here are different type of database we used like Ms Access, MSSQL, Oracle in source and destination. At present day there are only one language to other migration code conversion techniques. This paper attempts to give pros and cons of developed approach over existing ways in order to achieve enhanced database migration system.

Keywords: *Data migration, ETL, XML, Relational database, Schema-based, Mapping, Shredding.*

1. Introduction

As small scale organization gradually grows, need of expansion of database arises. Thus they have to switch to more efficient databases. Database translator can be used to translate data existing in a database to another database.

2. Literature survey

"To develop a Migration tool that helps in migrating database structures and data across various relational Databases. It will also facilitate migrating data from MSAccess, Excel Worksheet to SQLServer and MySql. It also migratesSQLServer - to - MySql and vice versa." The system will take an existing database in one format from the user and will convert it to a database in another format, which again will be specified by the user. Not only will it convert tables etc. The various other scopes can be enumerated as follows:

I. **Security**: The application is such that it segregates the authorized as well as unauthorized users.

II. **Time efficient**: Migrating every record manually would take quite a lot of time but using a migration tool would take the application few seconds to migrate from the specified source to the specified destination.

III. **Flexible**: The inherent nature of this application is flexible. The initial concept of data migration that we have implemented on typical case of Access to MySql can be further extended to any source database and to any destination database.

IV. **Interactive**: The highly interactive nature of this application ensures that minimum special training is required to handle it .

V. **Visually Attractive**: Interactive systems designed using Visual Studio makes the system visually attractive.

DESIGN DATA MIGRATION PROGRESS:

3. Development Methodology

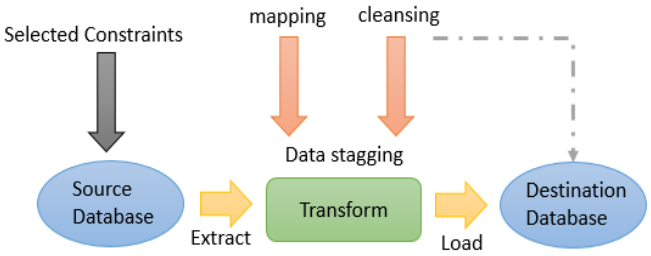

Figure1: *Complete block diagram of development methodology.*

3.1 Mapping

Mapping is the most crucial step for the success of any data migration. The mapping involved three steps

a) Mapping source tables.

b) Mapping source table to the physical data model (PDM).
c) Gap analysis to find if the entities mapped as present in existing customer Logical data model (LDM).

3.2 Design

Design included elaborating the various rules described in the mapping document. During this phase, it was required to revisit some of the mapping rules due to technical constraints. For transferring the files from source to the ETL environment, Oracle dump was extracted from the source system, transferred to the ETL environment and uploaded.

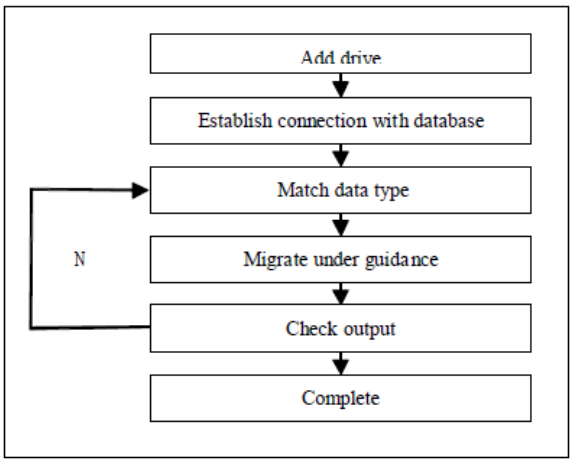

Figure 2: Flow chart of designing process of data migration.

3.3 Development

The development had two distinct stages:

i. Build scripts for new tables
ii. Build data transformation logic.

3.4 Testing

Two unique testing techniques followed by this programme helped earlier completion of testing cycle.

i. User testing before system integration testing –
In order to mitigate the risk of misinterpretation of transformation

ii. Tool based Regression Testing –
The frequency of change requests was very high which could impact the quality of deliverables easily. A regression tool which compared the data with previous delivery data and produces a report on mismatches automatically was developed. This helped to identify any regression issues created during these changes.

3.5 Cleansing

The data cleansing methodology we used to clean unnecessary data. While transferring data from one or source database to destination database some noise include which provide harm to current database so cleansing technique we use.

4. Difference between existing and proposed system

a. **Existing tool:**
1. Whole table is migrated.
2. Desktop migration
3. No time scheduling.
4. One way migration.
5. No user log maintenance.
6. Procedure, view, trigger we cannot execute in some cases.

b. Proposed tool:
1. Selective migration i.e. particular columns from a table can be migrated.
2. Web enabled migration. And desktop migration also included.
3. Scheduled based migration. So, it become easier to user for time schedule.
4. Both way migrations.
5. Maintaining user logs.
6. Procedure, view, trigger we can execute.

7. ALGORITHM : ODTDM - OBJECT DOCUMENT TYPE DEFFINATION MAP

Storing and querying XML documents using a RDBMS is a challenging problem since one needs to resolve the conflict between the hierarchical, ordered nature of the XML data model and the flat, unordered nature of the relational data model. This conflict can Be resolved by the following XML-to-Relational mappings: schema mapping, data mapping and query mapping. . In this paper, we propose:

(i) A lossless schema mapping algorithm to generate a database schema from a DTD, which makes several improvements over existing algorithms,

(ii) Two linear data mapping algorithms based on DOM and SAX, respectively, to map ordered XML data to relational data. To our best knowledge, there is no published linear schema-based data mapping algorithm for mapping ordered XML data to relational data.

Experimental results are presented to show that our algorithms are efficient and scalable.

1. Schema mapping,
Either a fixed generic database schema (schema-oblivious XML storage) is used, or a database schema is generated from an XML schema or DTD (schema-based XML storage) for the storage of XML documents. To support the ordered nature of the XML data model, an order encoding scheme such as those proposed in can be used and additional columns are introduced to store the ordinals of XML elements.

2. Data mapping,
Which shreds an input XML document into relational tuples and inserts them into the relational database whose schema is generated in the schema mapping phase.

3. Query mapping,
Which translates an XML query into its relational equivalent (i.e. SQL statements or relational algebra expressions),

executes them against the database and returns the query result to the user. If the query result is to be returned as XML documents, then a reconstruction algorithm is needed to reconstruct the XML subtrees rooted at the matching nodes.

The main contributions of this paper are:

1. We propose a schema mapping algorithm, ODTDMap, which generates a database schema from an XML DTD for storing and querying ordered XML documents. Although the main idea of ODTDMap is similar to the shared inlining algorithm and its variant, ODTDMap makes several improvements over them.

2. We propose an efficient DOM-based linear data mapping algorithm, OXInsert, which shreds and composes input XML documents into relational tuples and inserts them into the relational database according to the schema generated by ODTDMap. OXInsert is based on our previous data mapping algorithm XInsert, but it takes into account the ordered nature of the input XML documents and set-valued attributes that were not considered by XInsert.

3. We propose an efficient and linear SAX-based data mapping algorithm, SDM, which shreds and composes ordered XML documents into relational tuples and inserts them into the relational database according to the schema generated by ODTDMap.

4.1 Schema mapping algorithm ODTDMap :

In this section, we propose our schema mapping algorithm, ODTDMap, which generates a database schema from an XML DTD for storing and querying ordered XML documents. In ODTDM used to map a child and its parent to the same table when the child appears at most once under its parent. This operation is called in lining. The in lining approach reduces the number of tables in the generated database schema and thus the number of joins for a query. The ODTDMap algorithm consists of the following three main steps:

I. Simplifying DTD: Since a DTD expression might be very complex due to its hierarchical nesting capability, this step greatly simplifies the mapping procedure.

II. Creating and inlining DTD graph: We create the corresponding DTD graph based on the simplified DTD, and then inline as many descendant nodes as possible to a parent node in the DTDgraph. Thus, all descendants of an XML elements which occur at most once under e will be mapped to the same relation with e.

III. Generating database schema and s-mapping: After a DTD graph is in lined, we generate a database schema and s-mapping based on the inlined DTD graph.

IV. Data mapping: As the target database schema might be complex and its corresponding XML-to-Relational schema mapping is non-trivial, it is challenging to design an efficient schema-based data mapping algorithm.

The main challenging issues include the following:

a) **Varying document structure:** XML documents have varying structures due to the optional occurrence operators '?', '*', and choice operator 'j' used in the underlying DTD, unlike relational tables which always have a fixed structure. A data mapping algorithm should keep track of the missing child nodes and handle structural differences between the same type of element nodes due to the optional operators using efficient data structures.

b) **Scalability:** In an online environment, where new XML documents might be inserted into the database on-the-fly, a data mapping algorithm will be used frequently. Thus, it is critical that a data mapping algorithm is efficient and scales well with the size of XML documents. It is obvious that a linear data mapping algorithm will fulfill this requirement the best. In conclusion, the time complexity of OXInsert is O(n).

5. System Test after database migration

The system checks the quantity of migration, the results of which important criterions of deciding whether to start up new system.
The system test after migration includes:

- Completeness check: check the existence of reference of foreign key.
- Consistency check: make sure the same mining data have consistence value in bit.
- Records count check: check the consistency of records count in new system and in old system.
- Special sample data check: check the consistency of records count in new database and in old database.
- System integrity test: CPU speed, memory capacity, and migration time.

The integrity data contrast between old and new system to inquiry the same data by each inquiry tool and compare the result. First the data in new system are setback to what they are the day before migration in old system. Then patch the operation of the last day in old system into new one and compare the output. The system function test on windows 2000 and Windows XP. The result of system shows that connection between required database, mapping type and functions of data migration Simulated error condition indicate that system logs are correct and records of information are in time and accurate.

6. Conclusion

A "Database Migration Suite" is usually developed for individuals and organizations to save time for converting to a new database if a database already exists. The purpose of our

project is to migrate data from an existing database to another database.

Our migrator tool provides source and destination databases as Sql Server, My Sql, Oracle 10g. The very purpose of our project is to provide flexibility to client to migrate his existing database into a different database without any manual intervention.

7. References

[1] Bachman, C. W.: A Personal Chronicle - Creating Better Information Systems, with Some Guiding Principles. IEEE Transactions on Knowledge and Data Engineering, Vol. 1, No. 1, 1989, pp. 17-31.

[2] J. Zhiquan, L. Chengfei, S. Zhongxiu, Z. Xiaofang, C. Peipei, and G. Jianming, "Design and Implementation of a heterogeneous distributed database system," in Journal of Computer Science and Technology, published by Springer Boston, vol. 5, no. 4, pp. 363-373.

[3] P. Kokkinos, K. Christodoulopoulos, A. Kretsis, and E. Varvarigos, "Data Consolidation: A Task Scheduling and Data Migration Technique for Grid Networks," in Eighth IEEE International Symposium on Cluster Computing and the Grid, 2008.

[4] L. Golubchik, S. Khuller, Y. Kim, S. Shargorodskaya, and Y. J. Wan, "Data migration on parallel disks," in 12th Annual European Symposium on Algorithms (ESA), 2004.

[5] J. Hall, J. Hartline, A. Karlin, J. Saia, and J. Wilkes, "On algorithms for efficient data migration," in SODA, 2001, pp. 620–629.

[6] S. Khuller, Y. Kim, and Y.-C. Wan, "Algorithms for data migration with cloning," in 22nd ACM Symposium on Principles of Database Systems (PODS), 2003, pp. 27–36.

[7] S. Khuller, Y. Kim, and A. Malekian, "Improved algorithms for data migration," in APPROX, 2006.

[8] R. Gandhi, M. M. Halldorsson, G. Kortsarz, and H. Shachnai, "Improved results for data migration and open shop scheduling," in ICALP, 2004, pp. 658–669.

[9] Y. Pan, Y. Zhang, and K. Chiu, "Hybrid Parallelism for XML SAX Parseing," in 2008 ICWS '08. IEEE International Conference on Web Services, pp 505-512, 2008.

[10] Larson, J. A.: Bridging the Gap between Network and Relational Database Management Systems. IEEE Computer, Vol. 16, 1983, pp. 82-92.

[11] Rodgers, J.: Database Coexistence - Requirements and Strategies. Proc. of the 18th Mini-G.U.I.D.E. Conference, Florence 1989, pp. 117- 142.

[12] Lixian xing,Yanhong li,"Design And Application Of Data Migration In Heterogeneous Database," IEEE International Forum On Information Technology And Applications,2010.

[13] Jiahong Wang, Norihisa Segawa, Masatoshi Miyazaki," On-Line Data Migration Approaches and Their Performance Comparisons",IEEE Software and Information Science, 0-7803-7080-5/01, 2001.

[14] Chadi Kari, Yoo-Ah Kim, Alexander Russell, "Data Migration In Heterogeneous Storage System", IEEE International conference on Distributed Computing Systems, DOI 10.1109/ICDCS, 2011.