

# Assimilation of Various Databases into XML format

Dr G Arul Dalton

*Department of CSE,*

*MLR Institute of Technology, Hyderabad, India*

T Chandra Sekhara Reddy

*Department of CSE*

*MLR Institute of Technology, Hyderabad, India*

A Sravanthi Reddy

*Department of CSE*

*MLR Institute of Technology, Hyderabad, India*

Y Pavan Kumar

*Department of CSE*

*MLR Institute of Technology, Hyderabad, India*

**Abstract—** the ambition of this study is to discover a technique to assimilate several various databases like MySQL, Oracle, Sybase, Postgre, by taking the help of some of the modern Web technologies. It provides a global interface for these databases, accepts a query from the user in a natural language format (English), decodes this query into SQL using an interpreter, then retrieves data from the database, displays the result to the user and translates it into a fused format. XML is the fused format which helps to assimilate the data. To accomplish assimilation of data, we will implement an interpreter, which will take user query as its input and provides aimed appropriate data as output. The queries will be applied to core database to retrieve information from it and display to the end user. During this process, the retrieved data will be translated into XML format and piled as an XML document. The produced document can be used further for other XML processing like translating it into HTML or another XML document using XSL.

**Index Terms—**Assimilation, Database, Schema, System, XML.

## I. INTRODUCTION

XML is a very powerful mark-up language for data exchange over World Wide Web. Many are using XML for B2B transactions. Developers are updating methodologies to enhance and enrich the features of it. In this study, we put forward a practical framework for different database access. To develop this framework, we use XML schema as the canonical data model for a schema assimilation technique to enable the querying and assimilation of data from heterogeneous data sources. An important point of schema integration is to overcome the semantic dissimilarities of the schemas. If we have data source schemas in a recognized data model, we can provide users a single uniform system to facilitate data assimilation and querying without changing the core data sources. The proposed system will accelerate business integration by simplifying it dramatically - eliminating custom integration code, and replacing it with the straightforward assembly and configuration of off-the-shelf components. The system will provide everything necessary to achieve real-time, near-real-time, or batch integration more quickly and easily than competing tools and the 'roll-your-own' approach. It exploits XML schemas as the fusing data model for data assimilation and is depended on the mapping of the data source schemas into XML.

The main objective of this system is to access, combine, and reorganize Heterogeneous data that is locked away in legacy data silos. Data integration ensuring that information in multiple systems is kept consistent, this also known as heterogeneous information integration. Another objective is process integration it means linking business across applications. Common façade is also one of the objective for the Heterogeneous database

integration an EAI system could front end a cluster of applications, providing a single consistent access interface to these applications and shielding users from having to learn to interact with different software packages.

## II. EXISTING SYSTEM

The existing system consists of several users operating on different databases leading to multiple heterogeneous databases. Very often, data cooperation and interchange between different organizations is made difficult by the fact that little or no advance standardization exists and data is stored under different formats in distinct heterogeneous sources. Also some systems integrate the heterogeneous databases where data maintenance and retrieval becomes difficult.

### A. *Mermaid*

Mermaid is a system that allows the user of multiple databases stored under various relational DBMSs running on different machines to manipulate the data using a common language, either ARIEL or SQL. It makes the complexity of this distributed, heterogeneous data processing transparent to the user. In this paper, we describe the architecture, system control, user interface, language and schema translation, query optimization, and network operation of the Mermaid system. Future research issues are also addressed.

### B. *Pegasus*

Pegasus, a heterogeneous multi-database management system being developed by the Database Technology Department at Hewlett-Packard Laboratories, responds to the need for effective access and management of shared data across in a wide range of applications. Pegasus provides facilities for multi-database applications to access and manipulate multiple autonomous heterogeneous distributed object-oriented, relational, and other information systems through a uniform interface. It is not just a front-end approach to multiple databases but a complete data management system that integrates various native and local databases.

## III. PROPOSED SYSTEM

Hospitals need to exchange data among them to transfer the patient to a suitable hospital in a timely manner, to arrange appropriate treatment for the patient. Taking advantage of the Internet to exchange data can speed up the process of transfer of patient's data, it also evade the unnecessary recurrence of examinations and avoid the wastage of time and medical resources. To transfer the data, we must address the challenges of data transfer. The most important thing for data exchange is that the data format is common which will diminish the complexity of data interchange. The World Wide Web Consortium (W3C) proposed that the XML standard would be the solution to this challenge. In the past few years, XML has, marked as the standard for exchanging data on the Internet and on intranets. XML is playing a crucial role in the assimilation of various data. Starting out as a standard data exchange format for the Web, it has become active in the development of electronic commerce applications and online information services. As a results, the amount of XML data being stored and processed is large and will be increasing at a surprising rate. This has caused XML data management to become a focus of research efforts in the database community.

The goal is to assimilate data from existing system in a distributed environment while reducing the impact of operations on the databases. An approach is to use a fused global dissemination schema, such as the relational schema to provide effective global processing. Mermaid [10], Pegasus [7] and David [9] are all the examples of systems, which use this type of approach. These systems use schema integration methods but their global schemas become hard to manage as number and types of data sources increase.

The other approach for querying data sources involve a system based on Wrapper-mediator system. These systems are refined applications that abstract the data source from the users. Besides this, the system translates queries into the terms of the data sources and assimilate the results. SIMS [5] is an example for this system. The wrapper-mediator system is remarkably scalable, and allows the assimilation of huge number of various data sources. This research is similar to the work done with wrapper-mediators and global [2] schemas. This approach uses XML schemas as fusing data prototype.

## IV. SYSTEM ARCHITECTURE

The model of this system tries to simulate a public database system. This system is an interactive web application that is comprised of three modules User and an administrator and a query engine. The platform is based on ADL-SCORM standardization uses XML language to descript the resources and to validate resource description document by XML Data Definition Language (XML Schema). The resources are stored in a library of different kinds of resources, such as audio resource library, video resource library, PPT resource library. These libraries are stored in the diverse resources on the server. Bind a variety of learning resources metadata by XML

and then parse these metadata, so that map it which is currently the most popular net or standard metadata and is stored in the local server. In the resource description and management system, it has the various types of source and XSL template library. This library storage is applicable to various users and needs of template XSL file. The client module is responsible for loading the XSL template file of XSL template library, it provide to the users to personalize the user interface, respectively, and to achieve different functions.

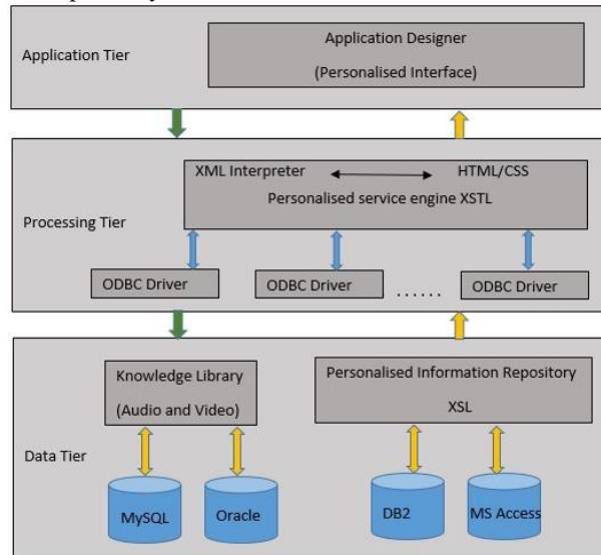


Fig. 1 The architecture diagram of various database assimilation platform.

This system architecture is systemized in three tiers, the User Tier, the Processing Tier and Data Tier (Fig. 1). Tier - 1 is the user interface built using jsp and html. Tier - 2, here the system communicates with both user and the data sources. The conversion of data which is stored in relational databases into XML tags happen here. The data sources are represented in the Tier- 3.

## V. EXPERIMENTAL RESULT

Once the data entered in the HTML form, when submitted it by clicking the Add Data button the data is inserted into the tables created in the database already. The same data which is inserted in the table is translated into XML format and the same data is stored in a document contains the data tags like below.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Emp>
<Empno>320581</Empno>
<Ename>Pavan</Ename>
<Job>Member Technical</Job>
<Salary>400000</Salary>
<Commision>20</Commision>
<Deptno>10</Deptno>
</Emp>
```

## VI. SCREENSHOTS

The results from this study are shown in the following screenshots with navigational steps.

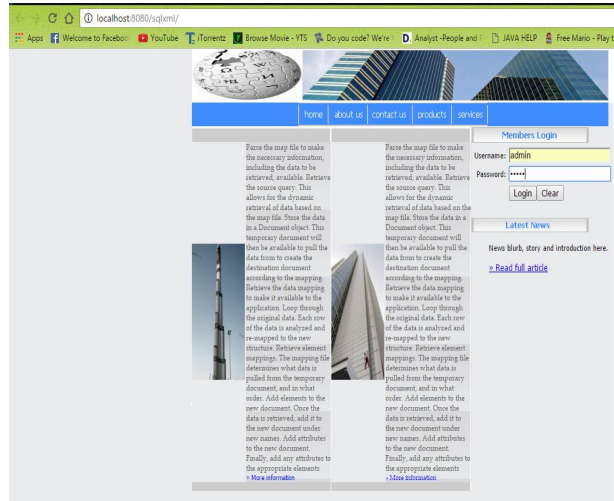


Fig. 2 Log In Page

Fig. 2 is the first screen appears when open the application where user can give the username and the password to login to application.



Fig. 3 Home Page

Fig. 3 shows the home page of the application where user could choose the options either Oracle to XML conversion, MySQL to XML conversion, Change Password or Log Out from the application.

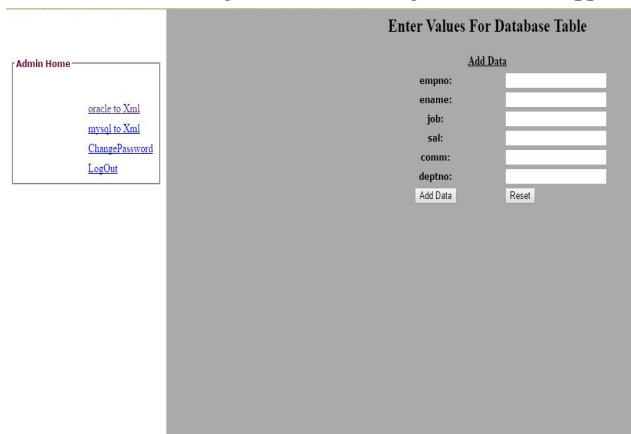


Fig. 4 Data Entry Page

Fig. 4 shows the Data Entry page where User could enter the data which inserts the data into a specified

database and generates an XML file.

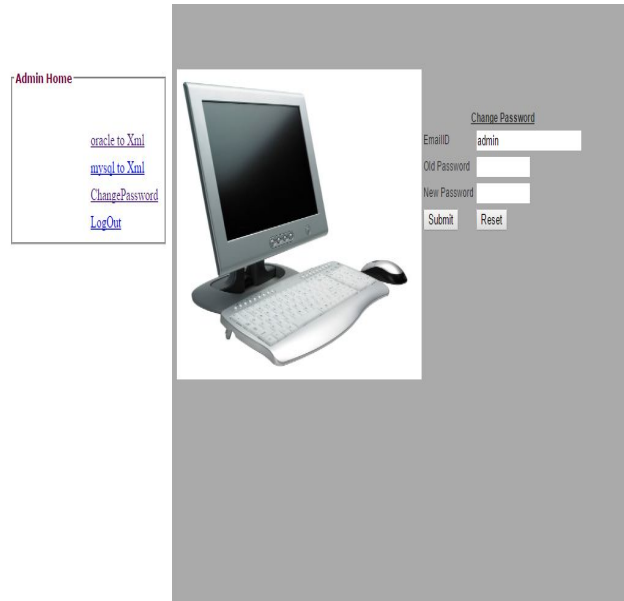


Fig. 5 Change Password Screen

Fig. 5 shows the Change password screen where user could change the password.

## VII. CONCLUSION

Heterogeneous Database Integration has inherent capabilities, openness and portability to heterogeneous environments makes them formidable competitors in their respective areas, but coupled together, provides an unbeatable, tightly integrated solutions. Heterogeneous database integration encompasses a range of techniques that all are coupled within Data Export, transformation & import, web services and messaging. Heterogeneous integration poses challenges beyond regular web application development. Successful Heterogeneous integration is scalable, standards complaint and cross platform. This is best platform to work on different databases with different environments. This system will be useful for working all type of database which can be integrated with each other.

## ACKNOWLEDGMENT

I would also like to show my gratitude to the HOD, Department of CSE and Principal, MLR Institute of Technology, Hyderabad for sharing their pearls of wisdom with us during the course of this research, and we thank reviewers for their so-called insights. Although any errors are our own and should not tarnish the reputations of these esteemed persons.

## REFERENCES

- [1]. Samuel Robert Collins, Shamkant Navathe and Leo Mark. "XML schema mappings for heterogeneous database access" <http://www.sciencedirect.com/science/article/pii/S0950584902000150>
- [2]. Frank S.C. Tseng and Chia-Wei Chen. "Integrating heterogeneous data warehouses using XML technologies" <http://jis.sagepub.com/content/31/3/209.short>
- [3]. Mohammed Adam Ibrahim Fakhraldien , Khalid Edris , Jasni Mohamed Zain , Norrozila Sulaiman. "Efficient Mapping XML DTD to Relational Database". [http://onlinepresent.org/proceedings/vol7\\_2012/4.pdf](http://onlinepresent.org/proceedings/vol7_2012/4.pdf)
- [4]. How to read XML file in Java – (DOM Parser) <https://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/>
- [5]. M. Ramanath, J. Freire, J. Haritsa, P. Roy 2003. Searching for efficient XML-torelational mappings, in: Proceedings of the International XML Database Symposium.
- [7]. R. Ahmed, P. D. Smedt, W. Du, W. Kent, M. A. Ketabchi, W. A. Litwin, A. Rafii, and M. Shan, The Pegasus Heterogeneous Multi database System
- [8]. Liu Sainan, Liu Caifeng, Guan Liming, 2009 Storage Method for XML Document based on Relational Database, In Proceeding of the IEEE International Symposium on computer Science and Computational Technology. Pages 127-131.
- [9]. B. Bhaskar, C. Egyhazy, K. Triantis, The architecture of a heterogeneous distributed database management system: The distributed access view integrated database (david), In proceedings of the 1992 ACM annual conference on communications 173-179
- [10]. M. Templeton, D.B. et al., Mermaid: A front end to distributed heterogeneous databases, In proceedings of the IEEE 75 (5) (1987) 695-708.