E-ISSN: 2278-179X

JECET; September 2013 – November 2013; Vol.2.No.4, 1381-1386.

Journal of Environmental Science, Computer Science and Engineering & Technology

An International Peer Review E-3 Journal of Sciences and Technology

Available online at www.jecet.org

Computer Science

Research Article

Data Integration for Distributed Databases

Ejiofor, V.E.[1], Nwachukwu, E.O.[2] and Enyinnaya, C. V.[3]

[1] Department of Computer Science, Nnamdi Azikiwe University, Awka-Nigeria

virguche2004@yahoo.com

[2]Department of Computer Science, University of PortHarcourt – Nigeria

enoch.nwachukwu@uniport.edu.ng

[3]Department of Computer Science, Abia State College of Health Technology, Aba,
myfathersblessing@yahoo.com

Abstract: This research focuses on data integration for distributed databases. The database represents the integration of the data/information of an organization. The data come from diverse sources which may have diverse formats. Data integration implies a well-organized effort to define and standardize all data elements to conform to a common format acceptable throughout the organization. The framework is implemented in the XML Web Service which does the data integration with minimal external guidance. Any request from any of the distributed databases is sent to the XML web service for harmonization, interpretation and signaling to the appropriate database that will handle the request. The reply is also communicated to the caller platform via the XML web service. The research methodology used is the Structured Systems Analysis and Design Methodology (SSADM). Object oriented methodology was also adopted in the design.

Key words: Integration, Distributed, Format, Standardize

INTRODUCTION

Data integration is the process of the standardization of data definitions and data structures by using a common conceptual schema across a collection of data sources. Data integration is the flexible and managed federation, exploration, and processing of data from many sources[1-5].

It is the use of common field definitions and codes across different parts of an organization. Data integration will increase along one or both of two dimensions:

1. The number of fields with common definitions and codes, or
2. The number of systems or databases adhering to these standards.

Data integration is an example of a highly formalized language for describing the events occurring in an organization's domain and its scope is the extent to which that formal language is used across multiple organizations or sub-units of the same organization[6-8]. According to Vincent Mastro[9], integration is the elusive goal of the software industry. It is the state that most software professionals pursue. The problem is that just few software are indeed fully integrated because of the difficulties in achieving true data integration. Vincent Mastro[9] stated that two systems are fully integrated when they

I. Look the same
II. Act the same
III. Consume and/or produce the same data.

The objective of data integration is to bring together data from multiple data sources that have relevant information contributing to the achievement of the users' goals. It is often required to integrate and analyze data from multiple sources. The major difficulty is that the data at different sources tend to be formatted in changing and incompatible ways, and even worse, represented under changing, incompatible and often implicit assumptions. For example, the customer databases of different organizations may use different formats for customer and agent names and addresses. Moreover, some data values in one schema may correspond to database or schema labels in another. Even worse, the same word may have a meaning, and the same meaning may have different names. This implies that syntactical data and meta-data cannot provide enough semantics for all potential integration purposes. As a result, the data integration process is often very labour-intensive and demands more computer expertise than most application users have. Therefore, a good data integration schema seems the most promising.

Also, it is increasingly important for organizations to achieve additional coordination of diverse computerized operations. To do so, it is necessary to have database systems that can operate over a distributed network and can encompass a heterogeneous mix of computers, operating systems, communication links, and local database management systems[11-13]. The system should be able to intelligently know where the particular data/information requested by the user resides in the distributed database. A real-time system[14-16] is considered intelligent when, with minimal external guidance it can perform complex actions in response to the sensed environment[17,18] (Payton D.W., 1992). Payton emphasized that a real-time system becomes more intelligent as it is given more capabilities to respond to its environment automatically, without detailed external guidance.

A database is a comprehensive, consistent, controlled and coordinated collection of structured data items. The database represents the integration of the data/information of an organization It consists of

logically related data stored in a single data repository. The database can be centralized in which case it is located in a single site or it can be distributed in which case it is located across several sites. The data derived from the entire organization and her customers must be integrated. Such data may come from multiple and diverse sources and have diverse formats. Data integration implies a well-organized effort to define and standardize all data elements to conform to a common format acceptable throughout the organization. Access to the accurate information in a timely manner is a significant challenge facing organizations today. The information are often scattered across numerous agencies. The problem, however, is that many organizations build information "silos" which are poorly accessible within their own organizations, let alone to the related departments outside the organization.

There has been a spectacular explosion in the quality of data available in electronic formats in the past few decades. This huge amount of data has been gathered, organized, and stored by a small number of individuals working for different organizations on varied problems. In light of the ever increasing volume of data, and the expected benefits of integrating the data, a framework for performing integration over multiple data sources for distributed databases[19-25] is necessary.

METHODOLOGIES AND TECHNIQUES

The methodologies used in this research are the Structured Systems Analysis[26] and Design Methodology (SSADM) and Object Oriented Methodology (OOM). In these methodologies, problems were identified, a feasibility study was undertaken, the present system was analyzed and the proposed system designed based on the problems identified in the present system. The program was coded and tested and the system implemented which must undergo a continual maintenance to keep the program in good working conditions, error free, and up to measurement. Object technologies were used to integrate specific data and the processes that create, read, update, and delete the data, into constructs called OBJECT. Once integrated, the only way to create, read, update, or delete the object's data (i.e. properties) is through one of its embedded processes (called methods). In order to achieve our objective of integrating data in a distributed environment, we divided the entire system into three different parts:

a. Front ends
b. XML Web service
c. Back ends (Databases)

Five different banks were used each representing a location of the distributed database. Data integration was achieved as every bank produces output of the same format despite the diverse forms/format of input.

OVERVIEW OF THE ENTIRE SYSTEM

The overview of the distributed system is illustrated in figure 1, which highlights the Bank administrator and customer, desktop and web application, XML web services, mobile phones, and the various distributed databases.

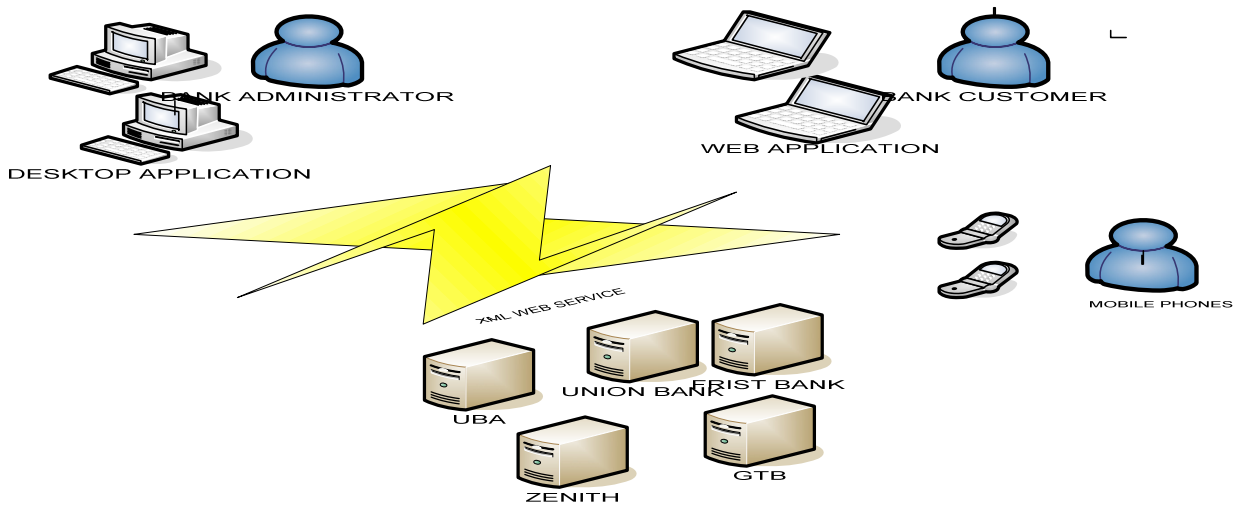

Fig 1: System Architecture

RESULTS AND DISCUSSION

The system can run on a desktop and on the internet. The system is designed in such a way that customers can check their account balance using their mobile phones and they need not come to the bank premises to do that. This ensures portability and convenience as the database can be accessed using a mobile phone. Customers can check their transaction details report and also transfer funds electronically using the web application. The system is as well able to manage integration of data across multiple databases in a distributed network with the XML web service. XML is a universal format that all platforms such as web applications, desktop applications, and mobile applications understand.

Organizations should, therefore, adopt the use of a distributed database management system to ensure the expected profit maximization. They should also move away from their heritage legacy systems and adopt an integrated platform where sources of data can be diverse and access to data can be made using different devices

CONCLUSION

While the 90s were about information, today is about answers. Answers drive our society and the ability to provide them quickly and cost effectively is a distinct competitive advantage. Today's high performance data centre needs to generate answers rapidly and requires that the right people can seamlessly access those answers. The ability to rapidly access stored information and answers is crucial to interpreting available information. This requires high-speed access and backup to a storage network.

Distributed data sources can be diverse in their formats, schema, quality, access mechanisms, ownership, access policies, and capabilities. Making the disparate pieces of the data centre infrastructure work together requires a high performance network. No longer simply providing transportation, but linking tens of thousands of servers, databases, and other computational resources, the network becomes involved in the creation of the answers. As such, it must be both fast and congestion-free to ensure that accurate answers are delivered rapidly.

REFERENCES

1. G.C.Barney, Intelligent Instrumentation – Microprocessor Applications in Measurement and Control, Prentice-hall of India Private Limited, 1988.
2. Bell & Crimson, Distributed Database Systems, Addison Wesley publishers, 1992.
3. Fitzgerald Jerry and Alan Dennis, Business Data Communications and Networking, John Willey & Sons. Inc,1996.
4. Haag Stephen, Maeve Cummings and James Dawkins, Management Information Systems for the Information Age, Irwin/McGraw-Hill publishers, 1998.
5. Ian Foster and Robert .L. Grossman, Data Integration in a Bandwidth-Rich World, Communications of the ACM, 2003, 46. No.11
6. C.J.Paul, A. Acharya,B. Black andJ.K. Strosnider, .Reducing Problem – Solving Variance to Improve Predictability, CACM, 1991, 34, 6, 80-93.
7. S.R.Schach, Classical and Object – Oriented Software Engineering, Irwin/McGraw-Hill1996.
8. William-Jan Van Den Heuvel and Zakaria Maamar, Intelligent Web Services, Communication of the ACM, October, 2003, 46, 10
9. Vincent Mastro, Data Integration – What Is It Anyway? Article published in DM Direct Newsletter March 5, 2004 Issue.
10. P.M.G.Apers, Data Allocation In Distributed Database Systems, September 1988 ACM Transactions on Database Systems (TODS), 1988, 13, 3.
11. A.P.Sheth and J.A.Larson, Federated Database Systems For Managing Distributed, Heterogeneous, And Autonomous Databases, September 1990 ACM Computing Surveys (CSUR), 1990, 22 3.
12. Ramakrishnan Raghu and Gehrke Johannes, Database Management Systems, McGraw-Hill, 2000.
13. Cruz .F. Isabel, A User-Centered Interface For Querying Distributed Multimedia Databases, Proceedings of the 1999 ACM SIGMOD international conference on Management of data table of contents1999,. 590-592.
14. Peter Rob, and Carlos Coronel, Database Systems, Design, Implementation, and Management, Course Technology, 1997.
15. J.A.Stankovic and K.Ramamritham, Advances in Real- Time Systems, IEEE Computer Society Press, Los Alamitos, California, 1993.
16. J.S.Lark,L.D. Erman,S Forrest,K.P. Gostelow,F. Hayes-Roth and D.M. Smith, Concepts, Methods, And Languages For Building Timely Intelligent Systems, Real-time Systems, 1990,2, 127-148.
17. D.W.Payton, Intelligent Real-Time Control of Robotic Vehicles, CACM, 1992, 34, 8, 48-63.
18. D.W.Patterson, Introduction to Artificial Intelligence and Expert Systems, Prentice-Hall of India Private Limited, New Delhi, 1992.
19. Carey Michael and Miron Livny, (1989), Parallelism and Concurrency Control Performance in Distributed Database Machines, June 1989 ACM SIGMOD Record, Proceedings of the 1989 ACM SIGMOD international conference in Management of data, 1989, 18, 2.
20. A.L. Powell, C.S. French, J. Callan, M. Connell and C.L. Viles, The Impact of Database Selection on Distributed Searching, July 2000 Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000.

21. F.A.Cavazos and L.J.Juan Carlos, Distributed Objects Research, Experience and Applications: A 3-tiered client – server distributed database system component-based, Proceedings of the winter international symposium on Information and Communication technologies (January 2004).
22. R.Mukkamala, S.C. Bruell and R.K. Shultz, Design Of Partially Replicated Distributed Database Systems : An Integrated Methodology, ACM SIGMETRICS Performance Evaluation Review, Proceedings of the 1988 ACM SIGMETRICS conference on Measurement and modeling of computer systems, 1988, 16 ,1.
23. K.Ramamritham, Real – Time Databases, Distributed and Parallel Databases, 1993, 1, 199-226.
24. G.Thomas, G. Thompson, C. Chung,E. Barkmeyer,F. Carter,M. Templeton,S. Fox and B.Hartman, Heterogeneous Distributed Database Systems For Production Use, September 1990 ACM Computing Surveys (CSUR),1990 , 22,3.
25. Knapp Edgar, Deadlock Detection In Distributed Databases, December 1987 ACM Computing Surveys (CSUR), 1987, 19, 4.
26. Jeffrey L. Whitten, Lonnie .D. Bentley, and Kevin .C. Dittman, Systems Analysis and Design Methods, McGraw publishers, 2001.

***Corresponding Author: V.E. Ejiofor;** Department of Computer Science, Nnamdi Azikiwe University, Awka-Nigeria