www.ijiarec.com

ISSN:2348-2079

DEC-2015

International Jmynal of Intellectual Advancements and Research in Engineering Computations

VIRTUALIZATION OF DISTIRIBUTED DATABASES USING XML

[1]M.Ramu

ABSTRACT

Objective is to develop a database virtualization technique for data analysts or other users who apply data mining methods to can use all ubiquitous databases in the Internet as if they recognized as a single database, thereby helping to reduce their workloads such as data collection from the Internet databases and data cleansing works. In this study, firstly i examine XML schema advantages and propose a database virtualization method by which such ubiquitous databases as relational databases, object-oriented databases, and XML databases are useful, as if they all behaved as a single database. Next, i show the method of virtualization of ubiquitous databases can describe ubiquitous database schema in a unified fashion using the XML schema. Moreover, it consists of a high-level concept of distributed database management of the same type and of different types, and also of a location transparency feature.

INTRODUCTION

Nowadays, massive amounts of data are collected daily in ubiquitous sensor network environments. With such data available and elaborately structured, it is more important than ever to locate and access knowledge and trends from it using data mining techniques.

A salient problem, hoiver, is that a person who engages in data mining using ubiquitous databases would have to spend much time for database selection and data collection. The primary objective of database virtualization is therefore to develop a technique so that the data analyst or other user can use all ubiquitous databases as if they ire recognized as a single database, thereby helping to reduce the user's workload.

XML TO RELATIONAL

As applications manipulate an increasing volume of XML data, there is a growing need for reliable systems to store and provide efficient access to these data. The use of relational database systems for this purpose has attracted considerable interest with a view to leveraging their poirful and reliable data management services. In order to store an XML document in a relational database, the tree-structure of the XML document must first be mapped into an equivalent, flat, relational schema. XML documents are then shredded and loaded into the mapped tables. Finally, at runtime, XML queries are translated into SQL, submitted to the RDMBS, and the results are then translated into XML. There is a rich literature addressing the issue of managing XML documents in relational back ends.

Several mapping strategies and query translation algorithms have been proposed. In addition, support for XML storage is already available in most commercial RDBMSs. Hoiver, none of these solutions addresses all the storage problems in a single framework. Works on mapping strategies often have little or no details about query translation and proposals for query translation often target a specific and fixed mapping strategy. In addition, many of the available mapping solutions hard-code mapping choices. A given mapping strategy is unlikely to be the best choice for all applications the ideal is to customize a mapping based on an application's characteristics, *i.e.*, its data and access patterns. Thus, hard-coding mapping choices can result in inefficient mappings. Although some of the solutions proposed by relational

Author for correspondence:

[1]Assistant Professor, S.Veerasamy Chettiar College Of Engineering and Technology, College Road, SVC Nagar, Puliyangudi- 627 855.

vendors do provide flexible mechanisms to define themappings, these solutions are proprietary and tied to a specific relational backend. This is a serious limitation. Since XML is widely used for data exchange, it is quite plausible that applications may need to store a given document or different views of document in distinct database back ends. Having to define distinct mappings, using different proprietary interfaces will be time-consuming and can add substantial costs to application development and maintenance.

INTEGRATION OF DATA

With the advances of Internet, the number of information smyces accessible through the Ib is increasing. Hoiver, these advances create new challenges. For example, there is a huge amount of related data made available by distributed providers. Rather than accessing and manipulating single database systems in isolation, research is needed to make it possible to simultaneously access and manipulate different remote databases. In addition to being distributed, the voluminous data are exposed by various data providers (e.g. institutions, organizations, companies, etc.), which have their own proprietary data models resulting in heterogeneity among databases. In order to provide transparent access to such remote data and enable the sharing of information among heterogeneous and autonomous databases, their schema heterogeneity needs to be identified and resolved. Proposing a solution for such problems is more challenging for environments whose members shall collaborate, while they pose a number of heterogeneities that need to be addressed by the infrastructure. For example, when a number of organizations are members of collaborative networks, the proposed infrastructure must support them with sharing and exchange of their information. More and more organizations understand the need to work together in order to better achieve their common goals.

The importance of collaboration has been ill understood in different domains, resulting in a rise in the number of collaborating organizations. It is important to provide an infrastructure enabling database interoperability, especially considering that collaborative networks need to be formed quickly. Heterogeneity is the most important obstacle facing the collaboration. Since data sharing constitutes the main type of collaboration, the collaboration infrastructure has to consider such differences for providing effective mechanisms to integrate or inter-link and homogeneously access heterogeneous databases. Hoiver, automatic resolution of schema heterogeneity still remains a major bottleneck for provision of integrated data access/sharing among autonomous, heterogeneous, and distributed databases. In order to provide transparent access to such remote data and enable the sharing of information among databases, their schema heterogeneity needs to be identified and resolved and then the correspondences among schemas need to be identified. This process is called as schema matching. After schema matching, schemas might need to be also integrated, depending on the needs of the collaborated systems.

SYSTEM ANALYSIS

Existing system

The existing system consists of the data virtualization. The virtualization of data is accomplished by using metadata, E-R diagram and UML. It is been used but not ill worked in all the platforms and also not suitable in all kinds of networks.

Drawbacks

- The use of metadata presents many advantages for creation that are irrelevant to what database model the metadata are based on. On the other hand, an important disadvantage is that they require a great workload to create them in their initial stage. Moreover, no definition and manipulation language to manage metadata has been standardized yet.
- The UML and the E–R model have similar fundamental characteristics; each has an advantage that its database design concept structure is irrelevant to what data model it is based on and with what DBMS product it is associated. Hoiver, those are only a few design techniques. No specific DBMS and definition manipulation language are provided.
- More workload is needed.
- Database management is found to be difficult.

The matters described in first two point's structured static schema for their use. Therefore, they have a difficulty in use with databases of various kinds that are available in the Internet in a flexible fashion.

PROPOSED SYSTEM

In my paper to achieve virtualization technique i use all ubiquitous data base as if they ire recognize as a single database, i will use an XML scheme that provides a flexible representation capability and a high transparency capability for ubiquitous data bases.

The XML scheme is now widely used to exchange information in the Internet environment. Database management systems are made than ever before; it is advantageous to use it because its definitions and manipulations are ill standardized. Hoiver, from a data model perspective, it presents the problem that it does not go ill with any object oriented data model that is now associated with multimedia. Even given that fact, now that the object-oriented concept is incorporated into the extended standardization of the SQL language, it has been improved in its affinity level with the associated XML scheme being converted to a relational database scheme.

In addition to that, the XML schema is a semi-structured dynamic one, it is still advantageous because of the fact that it is useful in a flexible fashion with various databases available on the Internet.

I proposed a means to recover the associated databases by allowing users to examine the virtual environment only for ubiquitous databases without having to examine real databases, and to ensure the integrity betien a virtual database and an associated real database.

The method of virtualization of ubiquitous databases proposed in my system describes ubiquitous database schema in a unified fashion using the XML schema.

Moreover, it consists of a high level concept of distributed database management of the same type and of different types, and also of a location transparency feature. Finally the common schema is generated.

FEASIBILITY STUDY

Economical feasibility

This is used to evaluate the technical aspects of the proposed system. This can be demonstrated if reliable hardware and software capable of meeting the needs of proposed system. It can be acquired or developed in the required time. This paper is technically feasible that satisfies the needs in the required time using the reliable hardware and software Economical Feasibility.

Operational Feasibility

This is frequently used for evaluating the operational feasibility of the paper. Only minimum number of operating persons is required, so the paper is operationally feasible.

Technical Feasibility

This is frequently used for evaluating the economic effectiveness of the system. The method is evaluated for the benefits and costs that are accepted from the system are considered and revieid. Further justification and alternation in the proposed system are incorporated.

Modules Description

It consists of fmy modules namely

- Virtualization of Database
- Common Schema Generation
- Data Conversion
- Query Conversion

Virtualization of Database

In this module the virtualization of modeled DBs of different types. For virtualization of different types of modeled DB, a single common schema is described for each model's schema information. The common schema used is an XML Schema. Around it, i will perform virtualization.

Common Schema Generation

The common schema provides the virtualized database structure for the application

programs; this schema is used to examine the syntax of query sentences and the constraints. The common schema generation program which converts the RDB schema into the common schema.

As a kind of special XML data, probabilistic XML has been also presented as the de facto standard for probability data exchange on the ib. And there are a mass of probabilistic relational data. In order to realize the conversion of probabilistic relational data to probabilistic XML data, the representation methods of probability data in the probabilistic relational data and in the probabilistic XML data are firstly analyzed separately. The probabilistic XML data tree is a classical probabilistic data model with some distribution nodes. The conversion process of probabilistic relational data to probabilistic XML data is composed of two stages what are schema conversion and data conversion

DATA CONVERSION

RDB data conversion into XML*:* The manner in which the RDB data are converted into XML. The XML tree structure, from unstructured content to XML and other structured information standards — quickly, accurately and cost-effectively. That's the goal of nearly every company about to embark on a major data conversion initiative and as the world's leading provider of data conversion services.

Benefits of XML data conversion over others are as

Xml is much Versatile Language than HTML. It Enable application that can't be done in HTML

HTML is quite limiting. It does not offer very rich semantics to describe a document. XML, on the other hand, allows ib designers and developers to build individualized, dynamic sites using complex data elements. It also enables them to access information across databases and types of data without having to rely on a search engine.

XML is robust, logically verifiable format is based on international standards.

Flexibility

XML offers very flexible options which are not possible in other languages.

Scalability

XML is scalable

- XML is platform-independent, thus relatively immune to changes in technology
- XML can be used across platforms: it is a standardized, vendor independent system for presenting data
- XML can be validated, that is, it must conform to a grammar or set of rules as presented in the Document Type Declaration (DTD). Such documents are easier to use and re-use than documents, which do not have a structured format for containing tags
- HTML can be easily converted to XML
- XML hierarchical structure is suitable for most types of document

Query conversion

To create the common query for retrieve the data from the various data base. The extension of the existing XQuery and the query conversion program from the XQuery language into SQL language or XQuery language

- XQuery is the language for querying XML data
- XQuery for XML is like SQL for databases
- XQuery is built on XPath expressions
- XQuery is supported by all major databases
- XQuery is a W3C Recommendation

Data Flow Diagram

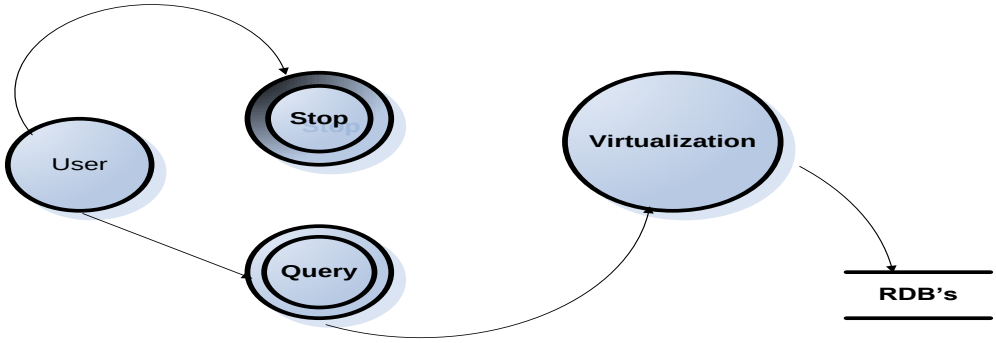

FIG: Virtualization

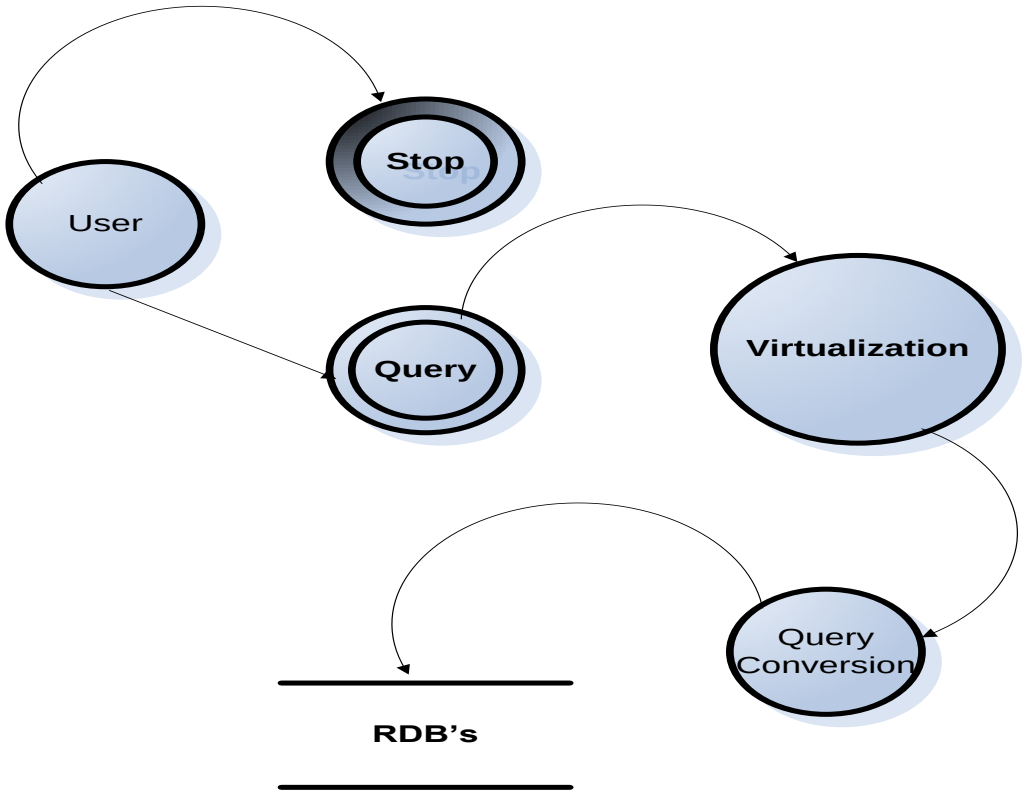

FIG: Query Conversion

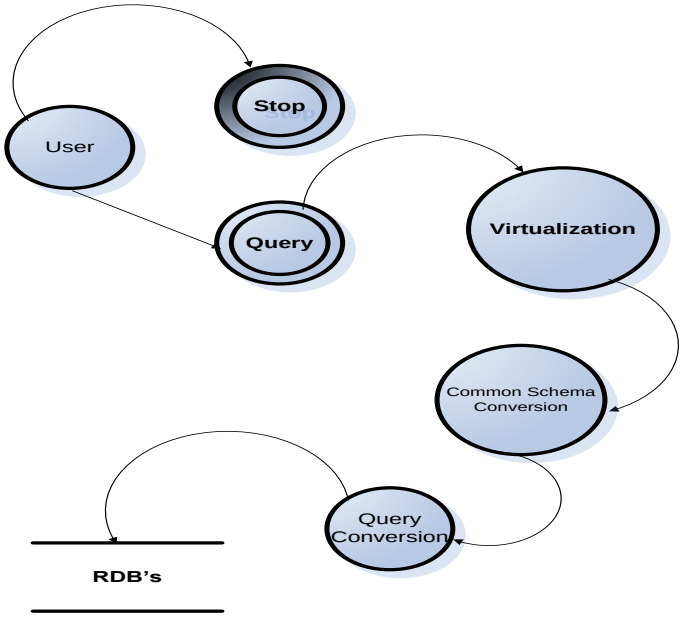

FIG: Common Schema Conversion

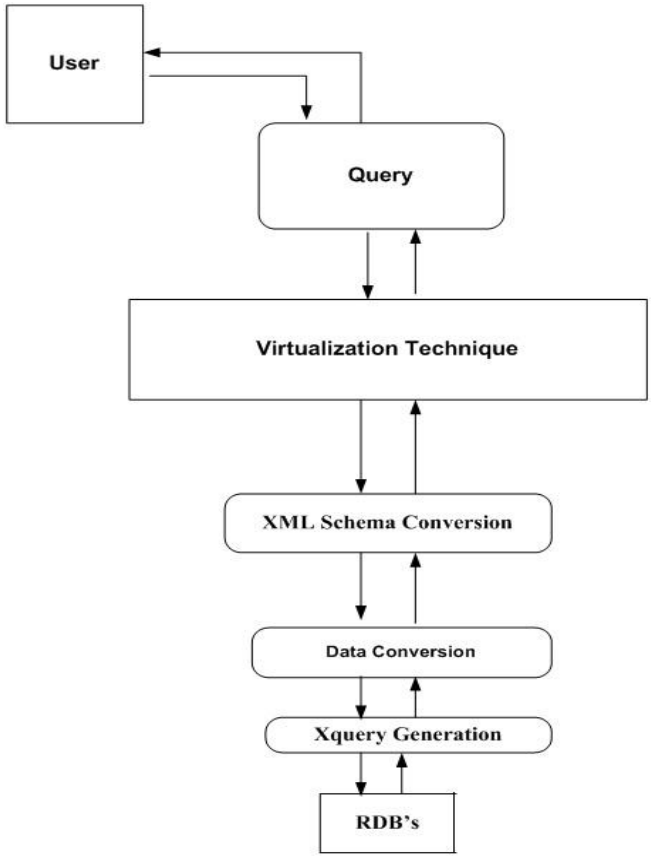

FIG: Overall Flow

SYSTEM IMPLEMENTATION

Implementation is one of the most important tasks in the paper. Implementation is the phase in which one has to be cautious, because all the efforts undertaken during the paper will be fruitful only if the software is properly implemented according to the plans made.

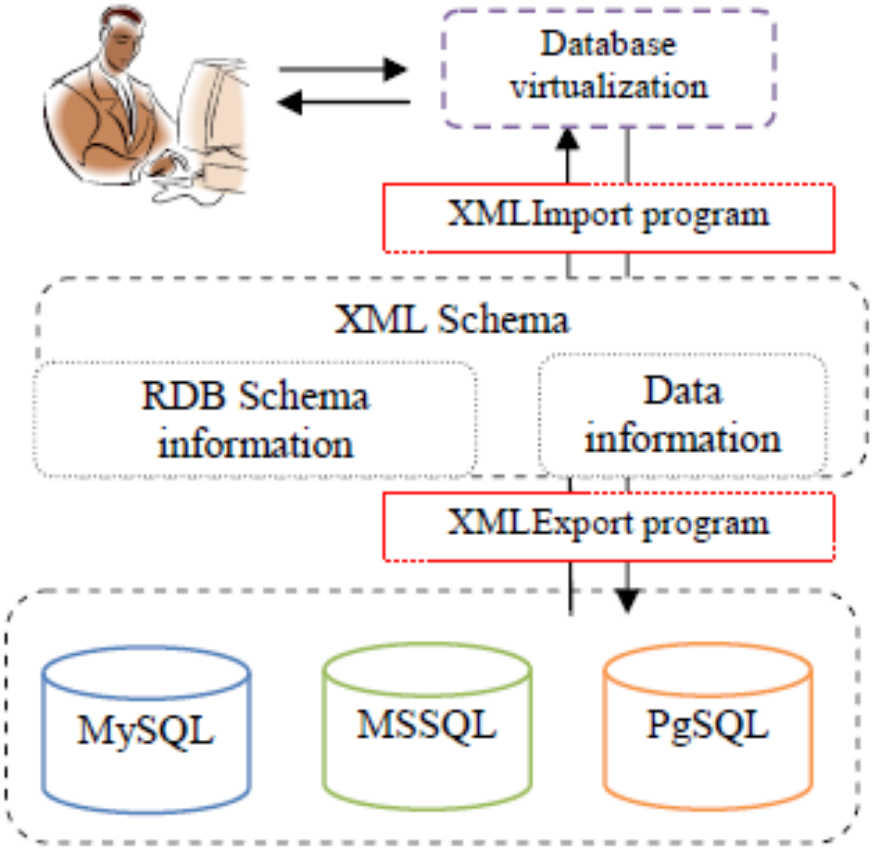

FIG: System Architecture

The paper is developed in java i.e. My Eclipse. The front end used for the paper is Java and the backend used for the paper is MYSQL server 2000. Hence i must have to install both this in the sever system i.e. in the ib server. The client does not need any software to be installed because it needs only the browser to interact; the browsers are installed in default when i install any operating system.

CONCLUSION

In this paper, i implemented the system that will be able to convert the Relational Database (RDB) schema into the equivalent XML schema through which adding a new value, updating a field, deleting a field are possible. So that the perfect utilization of database is obtained. Also the constraints of the table such as PRIMARY KEY, CHECK, NOT NULL, UPDATE CASCADE ON DELETE, UNIQUE can be converted without any problems.

FUTURE ENHANCEMENTS

In future, the XML schema could also be integrated to a common available schema. . In addition, i plan to implement the common data manipulation API (for example, extension of the existing XQuery modules) to access the virtual databases and i are going to incorporate location transparency functions to this API.

REFERENCES

[1]. STS Prasad and Anand Rajaraman 'Virtual Database Technology, XML, and the Evolution of the Ib', Junglee Corporation 1250 Oakmead Parkway, Suite 10 Sunnyvale, Ca 94086-4027.

[2]. Alexander Wohler and Peter Brezany and Ivan Janciak 'Virtualization of Heterogeneous Data Smyces for Grid Information Systems' Institute for Software Science, University of Vienna Liechtensteinstrasse.
Email:{brezany|janciak|woehrer}@par.univie.ac.in

[3]. Foster, Kesselman, C. Nick, J. And Tuecke,S. (July 2002) 'The Physiology of the grid: An open grid services architecture for distributed integrated systems.

[4]. Anand Rajaraman, Peter Norvig, (July/August, 1998) 'Virtual Database Technology: Transforming the Internet into a Database',IEEE Internet Computing, vol. 2, no. 4, pp. 55-58.

[5]. Abiteboul, S. Buneman, P. and Suciu, D. (1999) 'Data on the Ib: From Relations to Semi structured Data and XML', Morgan Kaufmann Series in Data Management Systems.

[6]. Varlamis, I. and Vazirgiannis, M. (2001) 'Bridging XML-schema and relational databases: a system for generating and manipulating relational databases using valid XML documents' Proc. ACM Symposium on Document engineering, pp.105-114.