J. Appl. Environ. Biol. Sci., 5(9S)646-652, 2015

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

Design and implemantation of xml based model for integration of distributed heterogeneous database

Tahere Kheradmandan[1], Sepideh Kheradmandan[2]

[1]Department of Computer, Sahneh Branch, Islamic Azad University, Sahneh, Iran
[2]Department of Computer, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran

Received: March 26, 2014
Accepted: May 17, 2015

ABSTRACT

As in today world most of databases are distributed and heterogeneous ,gaining this distribution and heterogeneity in different models is critical .one of the great challenges in the use of distributed database technologies is integrating object-oriented databases with relational database .on the other hand Maintain integrity and optimality in models is a very important and we can use the amount of Maintaining integrity and optimality as Criterion for comparing models.so first we review a model which works with different types of database and also is able to work disruption, and heterogeneity and then we offer a way to show the ability of proposed model and present a way to show the result of using this model can be very interesting . This paper first tries to review the model that could bring further coordination and integration between these two existing database models and then simulate this model and show the performance of the new model.

KEYWORDS: - Integration –optimality- model simulation – php code - distributed databases – heterogeneous databases

I. INTRODUCTION

Today, the ability to access information scattered in a heterogeneous and autonomous data sources, with the important needs of integrated approach is considered because the scattering position, the data is too scattered. Create interaction between the database, i.e., enable possibility of management and accessing to existing data in these databases in an integrated way So that the all user see the database system as an integrating system is important. So in this paper by using XML, the new model is presented, implemented and the results are shown.

II. DATABASE TYPES

A. *Relational database*

This model is a mathematical model related to concepts like predicate logic and the set theory. Data structure in these products is table with the difference that could have multiple rows. In other words there are multiple tables and the communication between them is not explicitly express Instead, the keys are used to matching those rows in different tables

B. *object-oriented database*

In this method, for each entity a class is created and each class has specific characteristics and properties then we introduce the elements as the member of classes. Object-oriented databases are systems based on the structure and concepts of multidimensional systems and to allow users to store objects directly in the database. Thus the structure of object-oriented programming can be used directly without converting to any other format.

C. *Distributed Databases*

A distributed database consists of sites with a loosely coupled connection that no physical component is shared. Each site can share the implementation of transactions for access to data is one or several sites

III. DESIRED DISTRIBUTED DATABASE STRUCTURE

One example of the structure of object-oriented, relational and distributed databases are as followed

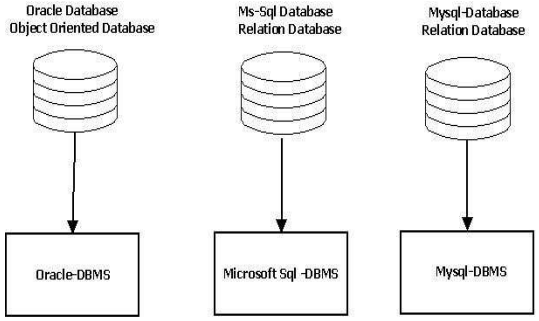


Figure 1. The discussed scenario with several heterogeneous databases of different DBMS

*** Corresponding Author:** Tahere Kheradmandan, Department of Computer, Sahneh Branch, Islamic Azad University, Sahneh, Iran

As it is clear in figure we assume a web server with three types of database ORACLE, MS-SQL, MYSQL .the type of ORACLE database is object-oriented and Microsoft SQL-Server and MYSQL database are relational.
In this example, the existing databases are different in kind of database and distributed property as well

A. ***Communication in distributed databaseDefine***

as the available database are with different DBMS so executive instructions are also different .the instructions must be presented to the database for operations on the DBMS via a SQL command interpreter engine or they must be translated or through the Gateway Such as (JDBC-ODBC-ADO-OLEDB) communication between the database and programs requests is establish . The advantage of using Gateway is that the program commands are translated for DBMS. In this case a processing stage is added to commands referencing to the DBMS but in this way different standard instructions problem for different DBMS is soluble

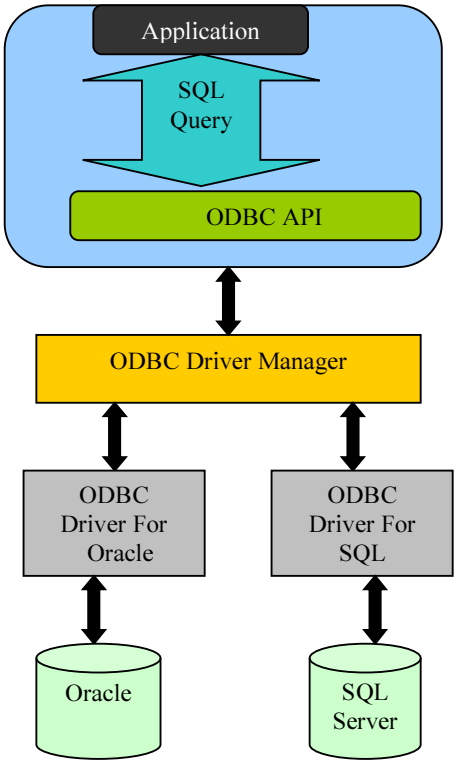


Figure 2. Implementation of instructions via interfaces to execute SQL instructions on various databases

The ODBC Weakness is that it works only with structured databases. OLEDB overcomes this weakness and Database can be a structured database or TXT files, or any other type of database. In OLEDB Access based on, Providers. Another issue that OLEDB solve is to support non-relational databases. As in the issue of connecting to databases by the use of Gateway, using the standard method which also could work with different types of databases (relational - object-oriented) is very important for coordinating relational database and object-oriented database. So defined gateways are based on OLEDB in order to be able to work with various databases. Connecting between programs and desired database apart from the type of database is by the use of drivers. In these systems, in Programming Layer there are 2 important Parameters:
1 - System sees all the databases as a local virtual unit database.
2- For connecting the database, in Program level transactions should not be involved with type of database and drivers call

So the key issue in distributed environments is to create a recognition engine for database type detection and diagnosis of type of connection. To keep these two characteristics a Middle engine for virtual database integration should be created to integrate databases virtually and sent Requests to the Desired Gateway. Fortunately, apart from the kind of a database (Relational - object-oriented) by converting to XML storage format it is possible to store various specifications of database with XML file thus the problem of database type is dissolved by using the interface engine and translation requests by the engine for Gateway. For maintaining performance optimality in system it is possible to manage the Gateway with a series of proposed traffic policy and the Gateway Performance kept fairly. Apply these policies; the Gateway s can be stored in normal condition. Traffic Control in Gateway will affected the whole system performance

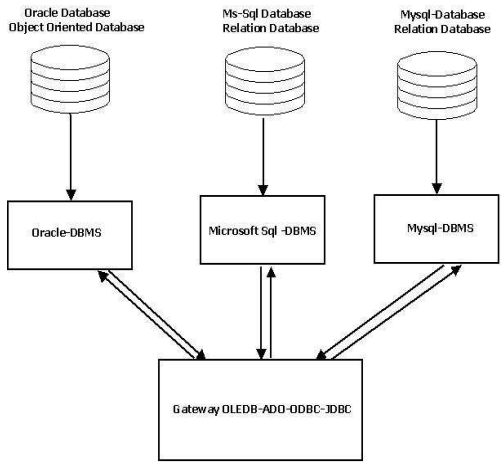


Figure 3. Perform the instruction via Gateway for performing SQL commands on various databases

B Table map of desired database:

For simulation we offer a data table which is used to work with the designed engine .we will describe this database later and now just show the table map.

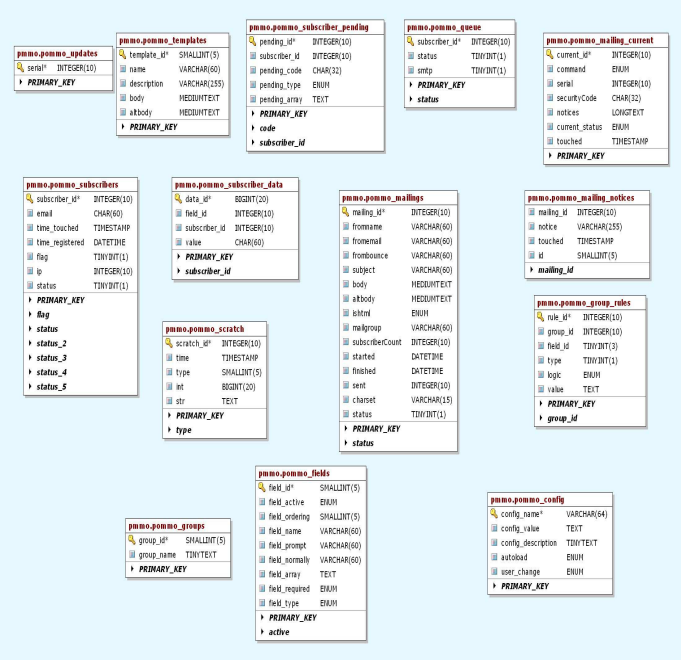


Figure 4. The table map of the simulated example

C PHP abilities

In The simplest level, PHP can do all work which is executable with the other CGI programs. Such as data collection from HTML forms, produce dynamic pages or send and receive Cookies. Perhaps the strongest and most important property of PHP, it supported many of the databases. Write a page related to the database to an incredible way, is simple. Now PHP supports protocols such as IMAP, SNMP, NNTP, POP3, and HTTP.

IV. PROPOSED MODEL

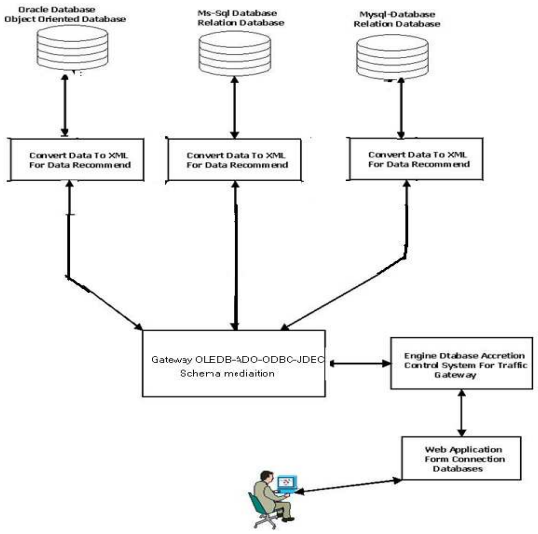


Figure 5. Proposed Model to coordinate the program with different types of databases (relational - object-oriented)

drivers which are in Gateway translate sql sent commands for desired database DBMS So identify the database and Gateway type can detect traffic created on the client side to manage the database. By this method request performance Efficiency and productivity on the database can be managed. This management can be a request process queue. This queue includes various commands to perform on distributed heterogeneous databases .in Figure 4 the model is shown. In this section describes the different parts of model

A. Types of database

Databases that are marked in the figure 4 have the variety of data storage and retrieval as well as the type of database and type of DBMS Hence, distribution and heterogeneous in the data banks are identified. Here we assume that the data are stored and retrieved in separate databases

B. Convert Data To Xml For Data Recommanded

in this Part a function is used to convert data. We Use this function to convert the data to standard XML format when they are stored, This method is suggested for database based on data because The magnitude stored or retrieved data structure is closer to the tree structure in productivity to store and retrieve XML format will have a better role This helped maintain structural condition of database as well as converting to the standard format for storing and retrieving data in the databases. On the other hand, if the data of an object-oriented database convert to XML format, then we can behave this database like a relational one which maintains an object-oriented structureFigures and Tables

In Figure 5 in DBMS part there are Various type of database management systems. The reason of using these types in this model is heterogeneity in different databases. Each database separately run the transaction with its own DBMS. So the heterogeneous issue is also considered By using the integration of heterogeneous databases as a virtual integrated database In this case, banks can be treated as a Centralized

V. SIMULATION AND RESULTS

For determining efficiency of used method we created a PHP page. This has the Ability to work with the requests send engine integrated. Main features of PHP (connect to different Gateway in different databases) cause to be able to connect to various types of DBMS in databases Engine is by using the designed engine all the tables inside the database and data of tables are converted to xml file

With this convert you can store all the data as XML .this Conversion is regardless of the type of distributed database. Using data conversion into the XML storage format, in addition to data storage can keep the relation between fields of databases as well. By the Presented way a comprehensive model for integrating relational databases and object-oriented is offered.

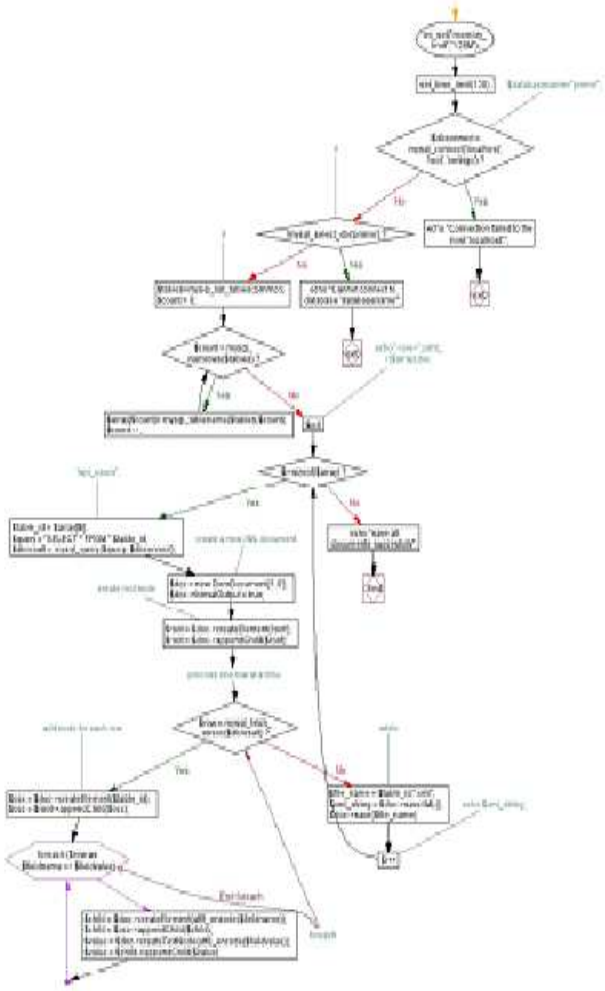
A. Implementation of banks and engine driver:

To implement the presented model first create MS-SQL, MYSQL and ORACLE databases. The banks are all expressing same reality but are defined

In various types of banks. The, table map has been presented before. After making the banks we should simulate different parts of the architecture.one of the most important of these steps is implementing the desired engine. We created a PHP engine with the ability to convert various data bases to xml storage format. To determine the effectiveness of this method we created a PHP engine which is capable of working with our application integratey. The main feature of PHP which is to connect to the Gateway of the different bank, has led to connect a variety of DBMS databases by using php.

For better understanding of the simulation model process the UML diagram and flowchart is drawn.in the depicted flowchart the algorithm is explained step by step. UML diagram uttered the states in the process.

Flowchart code:



Flow chart of mapping code to XML

In short this represents a conversion process and creates an XML document from a MYSQL table, the process is as follows:

First, in a predetermined time, by using a local number of banks we try to create a relationship with our bank. If we are not able to open a communication channel, with an error message the work ends. Otherwise, we try to find bank. If we become successful and the work continues, tables are recovered and converted to an xml document row by row. The diagram of the proposed algorithm shows to build an XML document from existing data banks. In the proposed method, after connecting to the bank and retrieve the specified table data is requested from each bank and each row in the table are converted to the appropriate tags.

B. ***Conclusion:***

By using the designed engine .First, all of tables in the database and data of tables are converted to xml file, with this conversion, all data can be stored as XML this means that such an approach can convert distributed banks to XML separated from its kind.

By converting the data to the XML storage format, In addition to data storage, we can also maintain the relationship between the fields of Bank. Because the XML feature is keeping means.

In this way we can provide a comprehensive model for integrating relational and object oriented data bases.

It essential to compare the way of working, the optimality and usefulness of the proposed approach with other existing methods

The bellow diagram shows the percent of integration by increasing the number of requests.in the low number of requests. The models are almost similar. But with the increasing number of requests Differences between models revealed as when the number of requests reaches to 1000, the proposed model is almost 25% more efficient.

Due to the simplicity of the concept and the ability to implement our model, this method can be easily used.

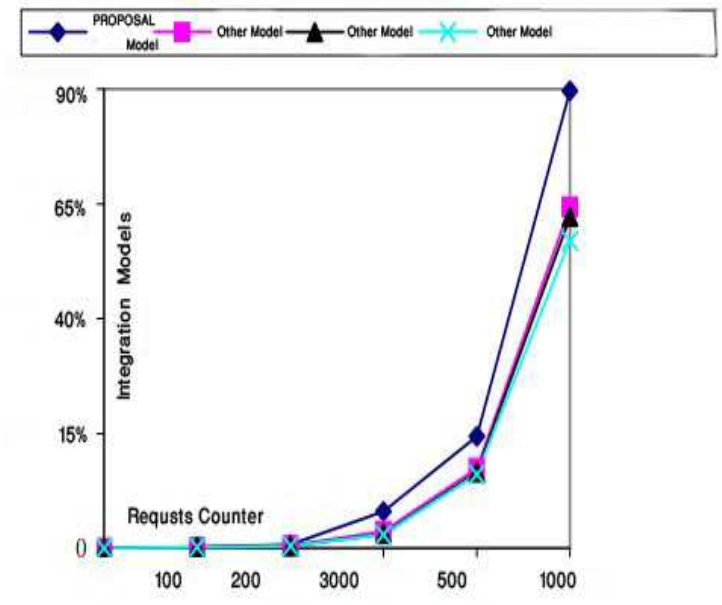


Figure 6. The comparison between presented model and other models

This figure compares the proposed method with three available methods.

This comparison is based on the speed of communication between different sites and getting an answer. This chart compares the optimality of integration with the increasing number of requests in different ways.

In this figure the blue curve shows the performance of the proposed model, the pink curve which has relatively a better result, shows the CORA performance in a simulated environment and black chart is a GSCW system, shows the performance of a mobile system. Features of this system are exactly the same as a distributed system and meet all the features of a distribution system. The only difference is, the different network communication protocols and these differences in protocols are not so important. The green curve shows the performance of a virtual machine.

Amount of Maintain the integrity and the optimality in presented models and the previous model is significantly.

REFERENCES

[1] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler. "Extensible Markup Language (XML) 1.0". World Wide WebConsortium (W3C). http://www.w3c.org/TR/REC-xml. Fourth Edition, 29 September 2006. (Visited on 2008-02-09).

[2] R.S. Sandhu, E. J. Coyne, H.L. Feinstein, and C.E. Youman. "Role-Based Access Control Models". IEEE Computer, Volume 29, No 2, pp.38-47, February 1996.

[3] L. Gong. "A Secure Identity-Based Capability System". Proc. IEEE Symposium on Security and Privacy, pp.56-65, 1989.

[4] J. Wang and S. L. Osborn, "A Role Based Approach to Access Control for XML Databases", SACMAT'04 of ACM, page 70-77, June 2004.

[5] E. Bertino, S. Castano, E. Ferrari, M. Mesiti, "Specifying and Enforcing Access Control Policies for XML Document Sources", World Wide Web, pp. 139-151, Vol. 3, 2000.

[6] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, "Controlling Access to XML Documents", In IEEE Internet Computing, pp. 18-28, Vol. 5, No. 6, 2001.

[7] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati "A Fine-Grained Access Control System for XML Documents". ACM TISSEC, pp.169-202, 2002.

[8] A. Gummadi, J. P. Yoon, B. Shah, V. Raghavan, "A bitmap-based access control for restricted views of XML documents", ACM Workshop on XML Security, pp. 60-68, October 2003.

[9] M. Kudo, J. Myllymaki, H. Pirahesh and N. Qi, "A Function-Based Access Control Model for XML Databases", CIKM'05 of ACM, page 115-122, November 2005.

[10] E. Funderburk, et al., "XTABLES: Bridging Relational Technology and XML", IBM system journal, vol. 41, number 4, pp: 616-641, 2002.

[11] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. "A model of authorization for next-generation database systems". ACM Trans Database Syst, 16(1):88–131, 1991.

[12] C. M. Ionita and S. L. Osborn. "Privilege administration for the role graph model". In Research Directions in Data and Applications Security, Proc. IFIP WG11.3 Working Conference on Database Security, Kluwer Academic Publishers, pages 15–25, 2003.

[13] S. Osborn and Y. Guo. "Modeling users in role-based access control". In Fifth ACM Workshop on Role-Based Access Control, pages 31–38, July 2000.
[14] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli. "Proposed NIST standard for role-based access control". ACM TISSEC, 4(3):224–275, 2001.
[15] E. Bertino and E. Ferrari. "Secure and selective dissemination of XML documents". ACM TISSEC, 5(3):290–331, 2002.
[16] T. Yu, D. Srivastava, L.V.S. Lakshmanan, and H.V. Jagadish, "Compressed Accessibility Map: Efficient Access Control for XML". VLDB, pp.478-489, 2002.
[17] M. Murata, A. Tozawa, M. Kudo and H. Satoshi, "XML Access Control Using Static Analysis". ACM CCS, 2003.
[18] C. Chan, Y. Ioannidis. “Bitmap Index Design and Evaluation”, In Proceedings of the International ACM SIGMOD Conference, pp. 355-366, 1998.
[19] J. Yoon, V. Raghavan, V. Chakilam, L. Kerschberg, “BitCube: A Three-Dimensional Bitmap Indexing for XML Documents”, In Journal of Intelligent Information Systems, pp. 241-254, Vol. 17, 2001.
[20] J. Yoon, "High Speed Access Control for XML Documents" IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 7, pp. 971-987, July 2006.
[21] V. Parmar, S. Hongchi, S Chen, “XML Access Control for Semantically Related XML Documents”, In Proceedings of the 36th Annual Hawaii International Conference, pp. 288–297, 2003.
[22] S. Boag, D. Chamberlin, M. F. Fernández, D. Florescu, J. Robie, J. Siméon, "XQuery 1.0: An XML Query Language", World Wide Web Consortium (W3C), http://www.w3.org/TR/xquery, January 2007. (Visited on 2008-03-17).
[23] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernández, M. Kay, J. Robie, J. Siméon, "XML Path language (XPath) 2.0", World Wide Web Consortium (W3C), http://www.w3.org/TR/xpath20 , January 2007. (Visited on 2008-03-17).
[24] T. Fiebig, S. Helmer, C. Kanne, G. Moerkotte, J. Nemnann, R. Schiele, T. Westmann, "Anatomy of a native XML base management system" , VLDB Journal, Vol. 11, Issue 4, Pages 292-314, Dec 2002.
[25] E.R. Harold, "Managing XML data: Native XML databases", IBM DeveloperWorks XML, http://www-128.ibm.com/developerworks , Jun 2005.
[26] X. Meng, D. Luo, J. Ou: "An Extended Role Based Access Control Method for XML Documents",Wuhan University Journal of Natural Sciences, Vol. 9, No. 5, pp. 740-744, September 2004.
[27] M. Tamer Özsu, Patrik Valduriez, "Principles of Distributed Database Systems", Prentice-Hall International, Upper Saddle River, N.J., 1999.
[28] R. Sandhu, E. Coyne, H. Feinstein, C. Youman. "Role-Based Access Control Models". IEEE Computer, 29:38–47, Feb. 1996.
[29] R. Sandhu, V. Bhamidipati, Q. Munawer. "The ARBAC97 model for role-based administration of roles". ACM Transaction on Information and Systems Security, 2(1):105–135, Feb. 1999.
[30] S. Osborn, R. Sandhu, Q. Munawer. "Configuring role-based access control to enforce mandatory and discretionary access control policies". ACM Trans. Information and System Security, 3(2):1–23, 2000.
[31] A. Gabillon, E. Bruno, "Regulating Access to XML Documents". Working Conference on Database and Application Security, pp.219-314, 2001.
[32] S. Cho, S. Amer-Yahia, L.V.S. Lakshmanan, D. Srivastava, "Optimizing the secure evaluation of twig queries". VLDB, pp.490-501, 2000.
[33] M. Kudo and S. Hada, "XML Document Security based on Provisional Authorization". ACM CCS, pp.87-96, 2000.
[34] A. Gabillon, M. Munier, JJ. Bascou, L. Gallon, E. Bruno. "An Access Control Model for Tree Data Structure". Proceedings of the 5th International Conference on Information Security, pp.117-135, October 2002.
[35] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, "Securing XML Documents", Proceedings of the 7th International Conference on Extending Database Technology (EDBT2000), pp.121-135, March 2000.
[36] D. Chamberlin, D. Florescu, J. Melton, J. Robie, J. Siméon, "XQuery Update Facility 1.0", World Wide Web Consortium (W3C), http://www.w3.org/TR/2008/CR-xquery-update-10-20080314, March 2008. (Visited on 2008-06-10).