

A Dynamic Replication Service for XML-Documents to E-commerce

A. Córdoba¹, J.J. Astrain¹, J.E. Armendariz¹, and J. Villadangos²

¹ Dep. de Matemática e Informática

² Dep. de Automática y Computación

Universidad Pública de Navarra

Campus de Arrosadía, E31006 Pamplona (Spain)

{alberto.cordoba, josej.astrain, enrique.armendariz, jesusv}@unavarra.es

Abstract. E-commerce transactions informations is traditionally stored in RDBMS. In this paper we consider that transaction tracking information is stored as XML documents in RDBMS geographically distributed. In order to support fault-tolerance and persistent storing service we propose a replication service. It provides a high available document retrieval against possible server failures, granting the access to the last version of the document. The service also allows to perform queries over the persistent data store in order to recover commercial transaction information from the replicated persistent store.

1 Introduction

E-commerce is one of the major concerns playing on the minds of corporate executives the world over. In interview after interview, senior managers say one of the biggest challenges facing their organizations is how effectively leverage the Internet, to give them a competitive advantage [1]. Each commercial transaction data of an e-commerce system is stored as XML documents, because this tag language is well suited to map it on Relational Database Management Systems (RDBMS) and to look for specific information in them. Tags are pattern elements that can be easily used to select the desired information, reducing the search space. Data in XML format is permeating all corners of the computing milieu: e-commerce, data streaming, web content, messaging, data interchange, database management. Many of these areas require breaking up XML into fragments and combining pieces of XML into larger documents, for purposes such as querying, stream monitoring, incremental updates and producing “recombinant documents”.

Many authors have persistently stored the XML schemas in *Database Management Systems* (DBMS). Many works can be found for RelationalDBMS [2,3,4] (mapping XML structures into tuples), Object-RelationalDBMS [5,6] (mapping XML structures into objects), applications in FederatedDBMS [7] and mixed solutions [8]. Our implementation uses as storage system a relational database as is depicted in [4].

The Object Group Service (OGS) [9] provides replication for CORBA applications through a set of CORBA services. Replica consistency is ensured through group communication based on a consensus algorithm implemented through CORBA service objects. OGS provides interfaces for detecting the liveness of objects, and mechanism for duplicate detection and suppression, and for the transfer of application-level state. This approach has been followed by different works, which uses the OGS or bases its group communication service on it [10,11,12,13]. However, this implies [14] that the Object Request Broker (ORB) should be modified. In [14], it is proposed an alternative based on an interception mechanism whose major advantage is that it does not require to modify the ORB.

We propose a CORBA application to provide high available XML documents containing e-commerce transactions. The replication service is based on the use of a replication manager and a set of geographically distributed repositories where documents are replicated.

The replication manager stores for each document its template (DTD), its current version and its repository owner. The replication manager is in charge of the correct replication of the commercial transactions and its management, that is, the creation, modification and deletion of XML documents stored in the repositories geographically distributed.

Each repository consists of an RDBMS and a local agent of the replication service, called Local Interface (LI), which translates XML documents to tables and viceversa. While there is at least one active repository, the replication service ensures that users get the last version of the document. It also supports failures at the local manager level and at the RDBMS.

The manager replicates asynchronously the XML documents. Each operation over a document is always performed firstly in the repository owner of the document. Once it is done, the manager updates asynchronously all the replicas in the remainder repositories. Moreover, the manager follows a passive replication policy. Whenever a document access fails, the manager receives an exception. This leads it to a state which, by means of its local information, tries to recover the document from other repository of the system. The system tolerates repository failures: LI and RDBMS stop failures.

2 System Model

E-commerce systems generate an important amount of relevant information that must be appropriately managed. Because of the relevance of the information generated by the business logic of the e-commerce system, it is usually saved in a persistent storage system. From the user point of view users interacts with the e-commerce system to do a commercial transaction (CT). The user connects to web services to perform a CT. These services provide the interface between the user and the e-commerce system. In commercial systems, different kind of CT can be issued. Each one contains a set of information that constitutes its template. Whenever a CT is performed, a set of information corresponding to its template is persistently stored by way of CTs and XML documents.

This paper proposes the use of XML schemas to store both CT information and CT templates. Each template has associated a XML schema. The use of XML schemas allows to store properly the information and to recover it from the storage system when needed. In order to perform the storage of the XML documents generated in each CT, the system implements a storing service that allows to keep the different XML documents according to their templates.

Commercial data and XML schemas are stored in a repository, each one is composed of an RDBMS (where data is stored) and its local interface (LI). The LI is the interface (by way of CORBA) between web services and RDBMS that allows applications to manage the commercial transaction documents. The LI receives storing/recovering requests of XML documents and translates locally these documents to a set of SQL sentences. An XML2SQL parser is used to do that, as it is shown in [15]. LI demands to the RDBMS the execution of the SQL sentences in a transactional way, which allows the LI to store/recover the requested information. It inserts a new database register when storing information and provides an XML document for the web service when recovering the information. The system stores/recovers XML documents that users/applications can update using web services. The LI component provides the translation from XML to a set of SQL sentences. These sentences, executed by the RDBMS, allow to store/recover XML persistent documents of an e-commerce system.

However, the system is prone to fail-stop or crash errors. In order to improve the fault-tolerance of the e-commerce system we propose and implement a replication service (RS). The RS is designed to ensure consistency and high availability of CT information by replicating repositories working in a replicate all policy. So, all available (non-faulty) repositories eventually will contain the same set of information. The RS is composed by two basic elements: the replication protocol (RP) and the information storage manager (ISM) that stores information processed by the protocol. RS interacts with the ORB agent and manages the replicas throughout all the Internet, being this process totally transparent to the users. LIs register to the RS in order to be considered for the replication process. The RS service can be implemented in anyone of the LIs that compose the system. The unique difference between an LI and the RS is the performance table employed by the RS to store information related to the performances and availability of the LIs composing the system. As the RS is the kernel of the proposed system, it is necessary to ensure fault-tolerance making use of UPS and RAID disks.

The RP protocol is composed by the system control protocol (SCP), and by the document storing protocol (DSP). The SCP establishes channel communications with the LIs of the system and detects failures in the replication system. The DSP protocol manages the storage of the XML documents into relational tables.

3 Conclusions

This paper introduces an on demand replication service, which allows the persistent storage of XML documents into RDBMSs, and their recovery whenever a repository crashes (LI or/and RDBMS). This service uses CORBA and is about

to be a liable alternative to the protocols supported by the CORBA standard, avoiding node grouping and multicast when connections are established in order to know the state of the system.

References

1. Reynolds, M.: Beginning e-commerce. Wrox Press (2000)
2. Kanne, C., Moerkotte, G.: Efficient Storage of XML Data. In Proceedings of the 16th International Conference on Data Engineering (2000)
3. Florescu, D., Kossmann, D.: Storing and Querying XML Data Using an RDBMS. *Data Engineering Bulletin*, 22(3) (1999)
4. Shanmugasundaram, J., Tufte, K., He, G., Zhang, C., DeWitt, D., Naughton, J.: Relational Databases for Querying XML Documents: Limitations and Opportunities. In Proceedings of the 25th VLDB Conference, Edinburgh, Scotland (1999)
5. Klettke, M., Meyer, H.: XML and Object-Relational Database Systems – Enhancing Structural Mappings Bases on Statistics. *Int. Workshop on the Web and Databases (WebDB)*, Dallas, (2000)
6. Shimura, T., Yoshikawa, M, Uemura, S.: Storage and Retrieval of XML Documents using Object- Relational Databases. In *Database and Expert Systems Applications*, Springer (1999)
7. Dittman,L, Frankhauser, P., Maric, A.: AmetaCar a Mediated eCommerce-Solution for the Used-Car-Market. In Ralf-Detlef Kutsche, Ulf Leser, and Johann Christoph Freytag, editors, 4. Works. “Federated Databases”, Berlin, Germany (1999) 18–33
8. Tufte, K., Maier, D.: Merge as a Lattice-Join of XML Documents. In Proceedings of the 28th VLDB Conference, Hong Kong, Chine (2002)
9. Felber, P., Guerraoui, R., Schiper, A.: The implementation of a CORBA object group service. *Theory and Practice of Object Systems*, 4(2) (1998), 93–105
10. Vaysburd, A., Birman, K. : The Maestro approach to building reliable interoperable distributed application with multiple execution styles. *Theory and Practice of Object Systems*, 4(2) 1998, 73–80
11. Cukier, M., Ren, J., Sabnis, C., Sanders, W. H., Bakken, D.E., Berman, M.E., Karr, D.A., Schantz, R. : AQuA: An adaptive achitecture that provides dependable distributed objects. *Proc. of the IEEE 17th Symposium on Reliable Distributed Systems* (1998), 245-253
12. Natarajan, B., Gokhale, A., Yajnik, S., Schmidt, D.C.: DOORS: Towards high performance fault tolerant CORBA. *Proc. of the International Symposium on Distributed Objects and Applications* (2000), 39–48
13. Marchetti, C., Mecella, M., Virgillito, A., Baldoni, R. : An interoperable replication logic for CORBA systems. *Proc. of the International Symposium on Distributed Objects and Applications* (2000), 7–16
14. Narasimhan, P., Moser, L.E., Melliar-Smith, P.M : Strongly consistent replication and recovery of fault-tolerant CORBA applications. *Journal of Computer Systems Science and Engineering*, (2002)
15. Keller, W.: Mapping object to tables: a pattern language (2000).
<http://ourworld.compuserve.com/homepages/WolfgangWKeller/>