

An Enhanced Secure and Scalable Model for Enterprise Applications using Automated Monitoring

¹A. Kannammal, ²V. Ramachandran, ³N.Ch.S.N. Iyengar

¹Department of Computer Technology & Applications, Coimbatore Institute of Technology, Coimbatore 641 014, TamilNadu, INDIA.

²Department of Computer Science and Engineering, College of Engineering, Guindy Anna University, Chennai 600 025, TamilNadu, INDIA

³School of Computing Sciences, Vellore Institute of Technology, Deemed University Vellore-632014, Tamilnadu, INDIA.

Abstract: E-Business must be highly secured and scalable to provide efficient services to millions of clients on the web. This paper proposes a new approach based on shared objects to improve security and mobile agents to improve scalability. The e-business uses shared objects and mobile agents to update the clients automatically with new information. The agent that resides in the database server is informed about the new information by triggering a function. Then the agent updates the shared object which is accessed by another agent that sends the information to the clients. This approach improves security, as clients are not aware of the location of central database and makes e-business more scalable by deploying mobile agents. The shared object is designed in such a way that it synchronizes the data transfer between agents. Proposed approach is implemented in a testing environment and the performance is analyzed. The analysis has shown that the proposed approach improves the security of business data and scalability of database servers providing synchronized data transfer.

Key words: Proxy Objects, Mobile Agents, Database Security, Scalability

INTRODUCTION

As electronic business expands all around the world, so do technical problems associated with it. Consequence of surges in volume of e-business delays transactions and outages occur as systems and networks become overloaded. Surveys and studies^[1] indicate that slow downloading time is one of the most often cited reasons that an on-line customer leaves the site and looks for another vendor's website instead. Since on-line companies' entire business depends on the behavior of their sites, long waiting time and unavailability can be disastrous. E-business sites try to follow the "eight-second rule"^[15], an unsubstantiated but widely held belief that after eight seconds of waiting for a web page to be downloaded, a customer becomes impatient and will likely to abandon the site. Therefore, for most electronic businesses, poor performance and low availability almost always mean lost revenue, bad press, bad public perception and a drop in the company's stock price.

Regardless of the type of e-business model adapted, there are some issues that have to be carefully

considered during design and implementation of an e-business site. These include the reliability, security, capacity, performance, scalability and cost of the system and network. E-business activities and Web Services are essentially real time processes in which performance, security and availability problems have a high cost. Frustrated users can translate into lost customers and lost revenue. Hence e-business must continuously guarantee quality of service to avoid losing sales and customers. Also, allowing users to directly access the business database may cause severe security problems. These problems demand a need for new approaches that would essentially improve the security and scalability of various servers present in the e-business environment.

This paper proposes a new approach based on shared object concept and mobile agents to improve the security and scalability of e-business applications. Shared objects and mobile agents can be used to notify the users on particular events such as insertion of new data or modification of existing data. When new data is created either by insertion or by updating, the agent that resides in the database server is informed about the

event along with the necessary information by triggering a function. Then it updates the users with up-to-date information through shared object and mobile agents. Further, the database is also secured by not allowing users to access it directly and by hiding the location of the database.

AGENTS IN E-BUSINESS

By increasing the degree of automation in e-business, it becomes much more dynamic, personalized and context sensitive and can be beneficial to the actors like buyers and sellers that are involved in e-commerce transactions. To achieve this automation and move to second-generation e-business applications, as software agents have become very popular in the web [2], a software model based upon the notion of interacting agents [3] is needed.

IDC, the premier global provider of market intelligence, advisory services, and events for the information technology and telecommunications industries [4] assumes that the dramatic growth in B2B e-commerce may accelerate the demand for agents. To enhance and improve the trading experience, agents can act as mediators in various processes of e-business. An agent can help the customer to recognize a need for some product. For this, it needs the user profile, which can be obtained in various ways like observing the user's behavior [5], through direct elicitation [6] or through inductive logic programming techniques. In Amazon Delivers [7], the latest reviews of exceptional new titles in categories that interest the user are sent automatically and Fastparts [8] uses "Autowatch" to allow users to list parts they need and notify them if those parts become available for sale.

With the expansion of the Internet, it is becoming more expensive and more difficult to navigate in order to find the necessary information on companies and their offerings. Given the difficulty and value, a common way of obtaining this information for companies in B2B e-commerce is through some form of information broker [9, 10] that acts as an intermediary between the buyers and sellers. Here a broker can be an agent or a multi-agent system. The functions offered by a broker may include information retrieval and processing, maintenance of a self-learning information repository about the user, profiling of users, monitoring for items of interest to the users, filtering of information, intelligent prediction of user requirements, commercial negotiation, collaboration and protection from intrusion [11]. Software agents [2] may be acting on customer's behalf to locate items in their profiles and find the sites that offer the best conditions in terms of

price, delivery time and return policy. A mechanism for replacing the work processes involved in various stages of e-business has also been proposed [12], in which a marketplace based on agents is developed. Though a variety of agent based deployments have already been made, there are a number of major research challenges that still need to be overcome before the full potential of agent-mediated e-commerce can be met [13].

The literature shows that much of previous work in this area focuses on the functional parts of e-business systems. It views the application of agents in this domain from either buyer's or seller's perspective. The human user's work processes are very much automated by using agent technology. In this paper a new architecture is proposed to use this agent technology along with shared object concept to improve the security and scalability of e-business systems.

PROPOSED SECURE E-BUSINESS SYSTEM ARCHITECTURE

The architecture of existing database servers in an e-business system is presented here by considering an example application that includes a database service provider (DBSP) which takes care of business database of its subscribers, one or many database owners who are the owners of the database and one or more database users who are interested in using the data present in the database. Figure 1(a) depicts the architecture of most common e-business application.

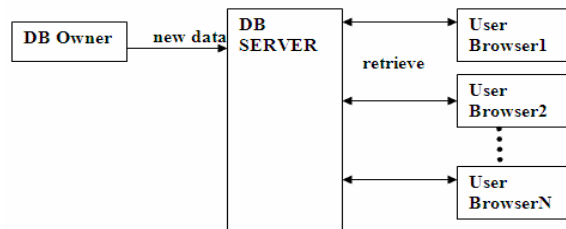


Fig. 1(a): Existing E-Business System Architecture

The DataBase owner is any business organization that wants to host its database in the DataBase Service Provider's DBServer (DBS). Here the users who are interested in the data have to continuously enquire DBS by sending requests. The DBS has to send responses to all the users. The users have to connect to the DBS all the time and continuously send requests.

The concept of Shared Object is introduced to improve the security of business database. Figure 1(b) shows the architecture of the system that includes the

Shared Object. When there is an information update, the DBS sends both old and new information to DataServlet through a trigger. The DataServlet checks the flag attribute of Shared Object to find whether the information provided by previous update has been consumed by WebServlet or not. If the flag is set to 'false', then DataServlet updates the old and new information attributes of Shared Object. Also, the DataServlet sets the flag attribute of Shared Object to 'true' to indicate that new information is available.

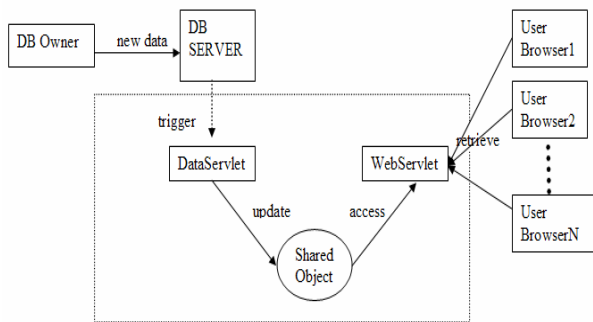


Fig. 1(b): System Architecture with Shared Object

The WebServlet which is the client interface continuously monitors the Shared Object for the availability of new information. When the flag attribute is 'true' then it retrieves the information and sets the flag to 'false' to indicate that the information is consumed. The flag field is updated accordingly in order to synchronize and reliable data transfer between the database server and the clients. The users have to contact the WebServlet continuously at periodic intervals by sending request messages for any information update. The users need not contact the DBS for the information. Also, the location of Shared Object and the DBS are hidden from the users. By this approach, the business database is made secure by avoiding direct access to the DBS. But the users have to send the request messages all the time continuously.

Agents are introduced in the architecture to improve the scalability of the system. Figure 1(c) shows the architecture that includes agents along with the shared object. Here, two stationary agents called InfoUpdaterAgent and InfoSenderAgent replace the DataServlet and WebServlet respectively. The shared object and the InfoSenderAgent are placed somewhere in the web such that the users are not aware of their location.

Whenever there is an information update, the database server sends this information to InfoUpdaterAgent that in turn updates the shared

object. The InfoSenderAgent continuously monitors the shared object for updated information. The users must have already registered with the DBSP for information update. On finding a new information, the InfoSenderAgent creates a mobile agent called InfoDispatcherAgent, provides it with the itinerary of list of users to be visited, and launches it. The InfoDispatcherAgent migrates to all users in the itinerary and provide them with the requested information. If they are off-line also, the agent updates them. Here, the users need not contact the DBS instead they have to register themselves with the DBServer for information update. Thereafter, they need not be in connect-state all the time.

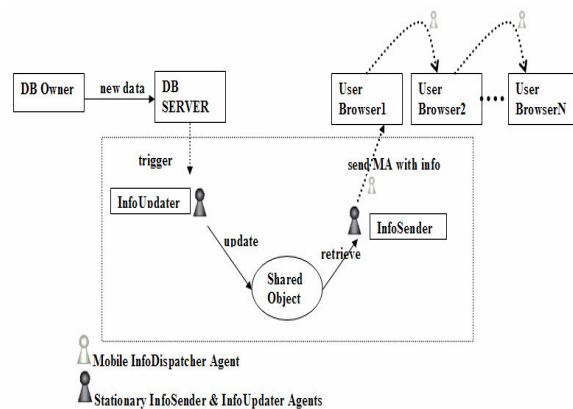


Fig. 1(c): Proposed E-Business System Architecture

By this approach, the database is made secure by avoiding direct contact of users to the DBServer and it is made scalable as large volume of users can be served. Further, the network resources are also conserved as the number of message exchanges is reduced. In the user's perspective, it saves the time and cost of connecting to the DataBase Server.

IMPLEMENTATION PROCEDURE

THE AGENT BASED INFORMATION UPDATING ALGORITHM

The algorithm to be executed by various components of proposed architecture is given below.

- Trigger
if (new data <> old data)
call function(new, old);
- Function
create InfoUpdate ;

provide info to InfoUpdater ;
start InfoUpdater ;

- InfoUpdater (UpdateAgent)
receive info;
identify SharedObject through
directoryServices;
while(old data not consumed)
 block itself;
 update SharedObject;
 block_on_event(acknowledgement);
 destroy itself;

- SharedObject
register with DirectoryServices ;
while()
 {
 if(request from UpdateAgent)
 {
 if(old data consumed)
 {
 receive data ;
 send acknowledgement ;
 }
 else
 block UpdateAgent ;
 }
 else
 if(request from MobileAgent)
 if(new data available)
 send data ;
 }
 }

- InfoSender (SenderAgent)
while()
 {
 If(new data available)
 {
 Create DispatcherAgent ;
 Provide itinerary ;
 Launch DispatcherAgent ;
 }
 }
 }

- DispatcherAgent
while(more sites to visit)
 {
 migrate to next site ;
 provide date ;
 }
 destroy itself ;

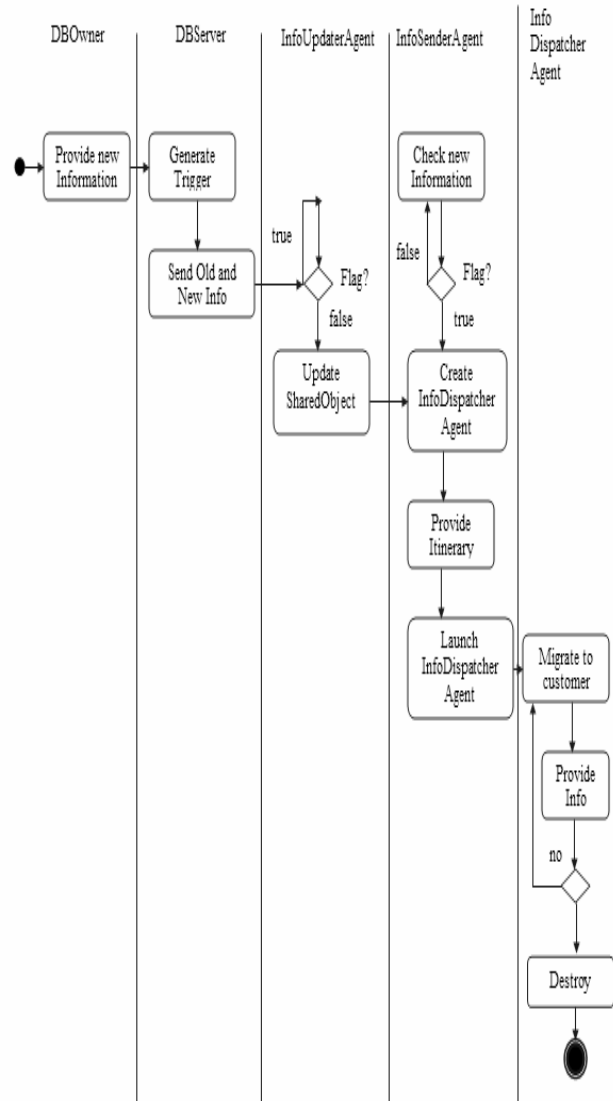


Fig. 2: Activity Diagram

PERFORMANCE ANALYSIS

ENHANCEMENT OVER TRADITIONAL E-BUSINESS SYSTEM

Enhancements offered by the proposed approach over the traditional e-business systems are :

- As in the traditional e-business system, the users need not be online continuously to get the required information update. Proposed approach eliminates this requirement and hence the cost and time components of the DB users have been saved significantly which in turn reduces the overhead involved in the clients' side.

The algorithm is well explained by means of the activity diagram as given in figure 2.

- The users are not allowed to access the database directly, and hence the users cannot execute unwanted SQL procedures in the database which in turn improves the security of business data.
- Even if the number of users who are accessing the database increases significantly, the DBS can handle them efficiently by deploying mobile agents as information providers, thereby improving the scalability of DBS.

Potential benefits associated with the proposed approach are described in detail in the following section.

SECURITY AND SCALABILITY IMPROVEMENTS

DBUsers once they register themselves need not contact the web server of DBSP for information retrieval. Hence the overhead involved in request response cycle is eliminated. Also, the network resources, web server session threads are also saved to a considerable amount as the DBUsers community is hundreds of times larger than the DBOwners community. Hence more and more users are encouraged to access the database while serving the existing users without any performance degradation. Even when there is a large surge of volume at an unexpected time, the DBS can withstand the situation.

The same approach can be used in a distributed environment, which is the natural scenario for many e-business applications. The database servers and the shared objects can be organized in a distributed manner, which implies even when there is a tremendous growth in the volume of users, the capacity of database servers can be increased either by using scaling out or scaling up approaches or directly by increasing the number of shared objects.

As the proposed approach hides the location of the database server from the users and eliminates the users accessing the database directly, the risk factor regarding security is also reduced considerably. Even the DBMS provides locking and access control rights mechanisms to ensure security, this approach opens a new window to security enhancements and it adds up the security of databases.

The updated data is immediately available in the shared object. The users when contact for the data will get it immediately even if the DBServer is busy serving others. Also from the user's perspective, they need not be on-line always to retrieve the information. They can be provided with the information even when they are

off-line by agents. This proves a considerable gain in the time and cost components.

The data in the shared object is updated only after the previous data is consumed. Also, the data is retrieved only if there is a new data. This ensures the synchronization of data exchange. In the earlier cases, if the user directly accesses the data, chances are there to retrieve old data or to miss some intermediate data between updates. These problems are avoided and the synchronization of data exchange is ensured at all the times.

CONCLUSION

In this paper, a new approach based on shared objects and agents is proposed. This approach enhances the security and scalability of e-business system. Architecture for the approach is designed and implementation details are given.

The proposed approach can be adapted to various applications in a generalized way. The approach considers all the data in the string form and can be used to deal with any kind of data. The data are considered as old and new to represent data before and after modification. The potential applications in which the proposed approach can be applied include: Home/Bank Security System in which alerts can be sent when there is an abnormal event like opening the door in the absence of owner, Share Price Updating System in which the interested shareholders can be informed of particular share prices reaching specified threshold values, Shopping Information System in which the customer can be notified when there is an offer for a particular product, Student Information System, in which the students can be notified about the events like results declaration, dates to remember, and so on.

ACKNOWLEDGEMENTS

We sincerely acknowledge the moral support and valuable suggestions offered by Dr. M. V. Krishnamurthy (Director, Academic Research, VIT), and the doctoral committee members Dr. S.K. Srivatsa, Dr. A. Kannan, Dr. Brijendra Singh and Dr.B.K.Ray during the entire course of this work.

REFERENCES

1. Delvin, B, Gray, J, Laing, B and Spic, G, 1999. Scalability Terminology: Farms, Clones, Partitions and Packs: RACS and RAPS : Microsoft Research Technical Report MS-TR-99-85.
2. Maes, P, Guttman, R and Moukas, A, 1999. Agents that Buy and Sell: Transforming Commerce as We Know It : Comm. ACM, vol. 42, no.3.

3. Jennings, N R, 2001. An Agent-Based Approach for Building Complex Software Systems : Comm of ACM, 44(4) pp35-41.
4. <http://www.idc.com>
5. Billsus, D and Pazzani, M, 1998. Learning collaborative Information Filters: Proc. of Fifteenth International conference on Machine Learning, pp 46-54.
6. Ribeiro, R A Fuzzy Multiple Attribute Decision Making: A Review and New Preference Elicitation Techniques, Fuzzy Sets and Systems, 78:155-181.
7. <http://www.amazon.com>
8. <http://www.Fastparts.com>
9. Ha, S H and Park, S C, 2001. Matching Buyers and Suppliers: An Intelligent Dynamic-Exchange Model: IEEE Intelligent E-Business, pp 28-40.
10. Gamvroulas, S, Polemi, D and Anagnostou, M, 2000. A Secure Brokerage Network for Retail Banking Services : Future Generation Computer Systems, 16:423-430.
11. Foss, J, 1998. Brokering the info-underworld : British Telecommunications Engineering, 17(2):202-206.
12. Ying-Hong Wang, Wen-Nan Wang, Chen-An Wang and An-Cheng Cheng, 2002. An Agent-Based e-marketplace over e-business : Proc. First Intl Symp. On Cyber Worlds (CE'02), IEEE.
13. Garfinkel, S, Spafford, G and Russell, D, Web Security, Privacy and Commerce, O'Reilly (2001).
14. <http://jade.tilab.com/>
15. http://www.businessweek.com/1999/99_30/c3639021.htm Trans. on Computer Systems, 9: 242-271.