December 2002

A Framework of an E-marketplace for Mobile Agents

Rohnan De Silva
University of New South Wales

Odysseas Sakellaridis
University of New South Wales

A Framework of an E-Marketplace for Mobile Agents

R. G. de Silva

School of Information Systems, Technology and Management
University of New South Wales, Australia
r.de-silva@unsw.edu.au

Abstract

High competitiveness and the growing popularity of e-Trading demand that the e-Commerce transactions are executed in a minimal time and cost. This paper proposes a new framework to enhance the inter-agent communication in e-Commerce transactions using the conventional mobile agents. This method performs the transactions at an e-Marketplace associated with a router, thus reducing the execution time and cost. The paper also proposes a strategy to group the agents into e-Marketplaces to improve the scalability of the solution.

Keywords

B2B and B2C trading, e-Commerce, mobile agents, router or gateway

INTRODUCTION

Merchants and customers rely more and more on automated forms of e-Commerce and as a result, agent technology is becoming increasingly important. Agents are software programs that can act on behalf of a user or an organisation in order to accomplish a task. There are two types of agents known as stationary and mobile agents. Stationary agents are executed only in the machine that they started the execution and may use remote procedure calls (RPC) to interact with the agents that are not on the same machine. In contrast, mobile agents are programs that can autonomously migrate through the network to a different location and perform tasks on behalf of their owners (Chih-Lin *et al.*, 2000). A mobile agent can pause the execution at one place, travel through the network to a different location and resume the execution at its new location. When mobile agents are used for e-Commerce transactions, they have to communicate with each other in order to exchange information, negotiate services or delegate tasks (Finin *et al.*, 1998). This is known as agent-to-agent communication or inter-agent communication. In the last few years, some agent frameworks, languages and protocols have been developed, especially for inter-agent communication in a mobile agent environment (Dasgupta *et al.*, 1999). While some of these approaches are language-specific, the others are more general in nature. Although these applications enable us to use an open standard, they will inevitably increase network traffic as well as response time. This will become more apparent when mobile agent technology will be widely used for e-Commerce transactions. Various attempts are made to reduce the traffic produced by inter-agent communications. Efficient dynamic routing (Tokumaru *et al.*, 2001) and active networked e-Marketplaces (de Silva *et al.*, 2001) are two such examples.

Traditional e-Marketplaces are vertical exchanges that focus only on specific types of goods and services (Feldman, 2000). Feldman quotes www.e-wood.com and www.e-metal-site.net as examples of such vertical e-Marketplaces. The users of these sites, whether they are buyers or sellers, have to manually enter the product or services that they want to purchase or they have on offer using their web browsers. As such, these e-Marketplaces do not offer a fully automated negotiation and settlement environment. The owners of these sites are brokers and they facilitate the successful negotiation and settlement of transactions between the buyers and the sellers. The e-Commerce transactions between a merchant and a customer can be automated using mobile agents. Lange and Oshima (1998) define an electronic marketplace as a multi-agent system in which selling and buying agents interact. They provide TabiCan as an example of such an e-Marketplace. Tabican is an e-Market place for air tickets and package tours and can host thousands of agents. In the server of the TabiCan, shop agents wait for requests from consumer agents. Users can leave behind their mobile agents for up to 24 hours in the e-Marketplace. These agents have the autonomous decision making and mobility capabilities. They can monitor trading opportunities, search for trading partners and products, and make decisions to satisfy the objectives, preferences and constraints of the owner according to his or her trading rules. The participants of the e-Marketplace are the marketplace owners, shop owners, and the consumers. The marketplace owners have the role of managing the system resources such as hardware and databases. Shop owners and the consumers are the users of the e-Marketplace. This is the second and emerging type of e-Marketplace. The type of trading that occurs in both these types of e-Marketplaces could be either business-to-business (B2B) or business to consumer (B2C). The major disadvantage of this second approach of e-Marketplaces is that there will be a large number of e-Marketplaces and as a result a large number of customers' agents will roam around the Internet for shopping. Since each e-Marketplace send market advertisers (Lange and Oshima, 1998) to other e-Marketplaces to attract consumers, number of agents in the Internet will increase further. This will inevitably increase the response time. A second problem is that they are still vertical marketplaces where only a single product or service is available. Extension of such an e-Marketplace to a horizontal e-Marketplace that will have a large number of products and services in offer will not be straightforward. Because of these reasons, the physical locations of the e-Marketplaces and the method of agent transfer among them have to be specified to minimise these problems.

We have recognised two solutions to the problems mentioned above; the first is to use an active router instead of a legacy router and to create the platform for e-Marketplace on it. This approach has been introduced and discussed in a recent paper (de Silva *et al*., 2001). For the successful operation of this method, however, we needed to develop an Agent Communication Protocol (ACP) in the network layer. We have already developed the first version of this protocol (de Silva *et al*., 2002) for the communication of active agents. However, for the successful implementation of this type of e-Marketplaces, we need to have active routers and active networks in place. The second approach, which is discussed in this paper, is to retain legacy routers but to attach a server and a database to the router, which when combined with the router, provides a platform for the e-Marketplace. We call this router system an E-Marketplace Router System (ERS). This method has the advantage that we can use a standard Agent Communication Language (ACL) directly without using special protocols for network level communication. However, this design should address the issue of improving efficiency and reducing latency. As in the case of an e-Marketplace located in the active router, we can have two physical locations for the negotiation between the merchants' agents and the customers' agents: In e-Marketplace scenario, the information delivered by the merchants' agents will be stored in the ERS and the mobile agents of the customers will visit these systems to find a suitable matching. The router in this case, is a core router owned by an Internet Service Provider (ISP). In the e-door-to-door selling scenario, mobile agents of the customers will store the information in an ERS located in their home network, and the merchants' agents will visit these routers in order to find a prospective customer. The router in the latter case is usually an edge router of the organisation. We do not discuss this scenario in this paper, as it can be easily adapted from the former scenario.

The rest of the paper is organised as follows. In the next section, we provide a brief description of the existing forms of inter-agent communication. In the following section, we present our proposed method. The method for the allocation of the agents to ERS is presented in the succeeding section, which is followed by the section on conclusion.

BACKGROUND

In this section, we will examine some existing forms of inter-agent communication and discuss the advantages and disadvantages of them. Traditionally, multi-agent systems have used Agent Name Servers (ANS) to enable interaction between static agents. In the case of Internet-based systems, agents use ANS simply to look up the IP address of another agent and then to use that address to make a socket connection directly to that agent for the purpose of exchanging messages. The problem with this approach is that if the IP address of the latter changes, the former will only find it out when the next attempt to send a message fails. Also, if an agent crashes due to some reason, it is the responsibility of the other agents, with whom the former was communicating, to properly save any failed messages and re-transmit them later. Improved approaches include the JATLite Agent

Message Router (AMR) (CDR, 1998) and the middle agent entity. AMR is a special application that receives messages from registered agents, and queues these messages in the file system before routing them to the correct receivers. Middle agents are entities to which other agents advertise their capabilities, and are neither requesters nor providers from the standpoint of the transaction under consideration. Their operations are similar to AMR, but they can perform other important functions like acting as facilitators and mediators. On the other hand, as mentioned previously, mobile agents are programs that can autonomously migrate through the network to a different location and perform tasks on behalf of its owner. It can pause the execution at one place, travel through the network to a new location and resume the execution at its new location.

The forms of inter-agent communications we discussed above involve an Inter-agent Communication Language (ACL) such as Knowledge Query and Manipulation Language (KQM), Foundation for Intelligent Physical Agents (FIPA) ACL (Breugst *et al.*, 1998) that provides a tool and a framework to handle the interoperability problems of inter-agent communication of agents. However, so far, there is no common standard developed on which the agent communications can or should be based. Figure 1 shows an example of the inter-agent communication scenario between two mobile agents. The mobile agent is a program that is initially located in its owner's computer. This is called the default location of the mobile agent.

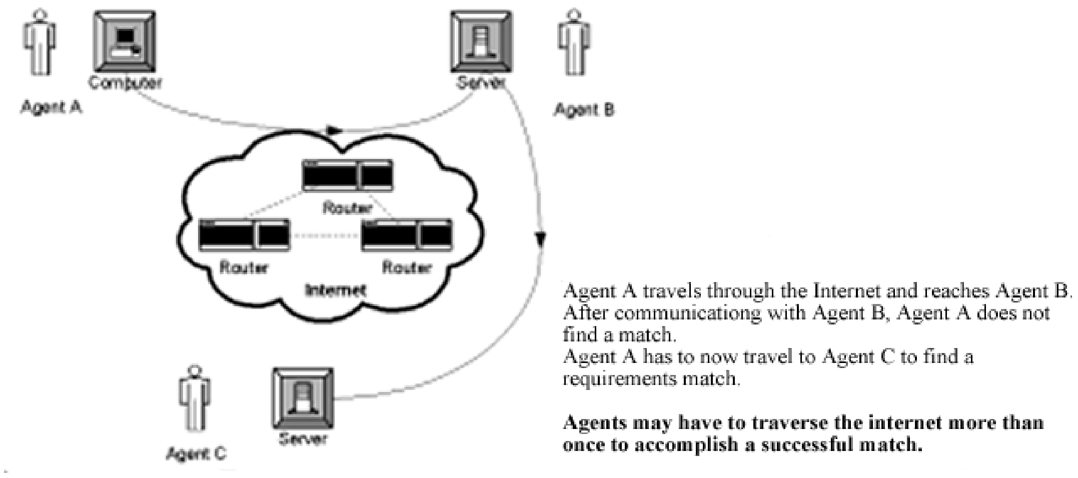

Figure 1: A typical scenario of inter-agent communication

The standard inter-agent communication model using mobile agents which involves communication between two processes of the two application layers in the Open Systems Interconnection (OSI) model requires the customer's agent to visit the merchant's agent at the merchant's site. If this visit does not result in a successful negotiation, the customer's agent has to try other merchant's agents. This will sometimes mean traversing the entire Internet from one end to the other several times. Though the mobile agent-based e-Marketplaces that was discussed previously can reduce the number of such travels initially, as the number of competitive broker sites for the same product or service grows, several visits to many broker sites by a buyer's mobile agent would be inevitable.

PROPOSED APPROACH

In the proposed method, an ISP owns the ERS and the stakeholders of the ERS are a number of prospective buyers, sellers or brokers. These stakeholders have the applications that are capable of performing the transactions using the ERS and their mobile agents. To be able to use the services provided by the ERS, each user should register with their respective ISP. ISP provides the service contract for the user that specifies a certain level of Quality of Service (QoS) and the IP address of the core router of the ERS. Depending on the service contract, the ISP provides the user with membership of other ERS systems owned by other ISPs. The service contract may define fixed cost for using a set of ERSs or a differentiated service such as a different cost for each ERS in the ERS set. This enables the user to participate in trading in the designated ERS systems.

We now describe how the transaction takes place referring to Figure 2 and using a simple example of purchasing a medication. Initially, the merchants B and C will send their mobile agents with the information about their available medications to the core routers in their respective ERSs (not shown in the figure). Once these packets arrive at each router, they will be directed to the respective server. After being directed to the server, the information of the medications and that of the agents are stored in the database. Depending on the service contract, this information is shared between several other ERS systems. Let us assume that this information was shared and is available in the ERS that is shown in Figure 2. If the customer A wants to purchase a certain medication, he or she instructs the Agent A about the required specification of the product. As Customer A is registered with the ISP to use this ERS, the mobile agent Agent A will now travel to the ERS where it is directed to the server of the ERS. The execution environment in the server makes the inter-agent communication between the merchants' agents and the customer's agent to take place. Let us assume that after a successful negotiation, a match was found with the product of Merchant C. Now, Agent A as well as Agent C will travel to their default locations and inform about the successful negotiation to their owners Customer A and Merchant C respectively. In this scenario, the ERS will have all the information about the merchant's agents and rapid communication and negotiation is made possible. If the Agent A could not make a successful negotiation, it has to travel to the next ERS system. For this purpose, the ERS provides the customer's agent with a list of ERS systems that it should visit.

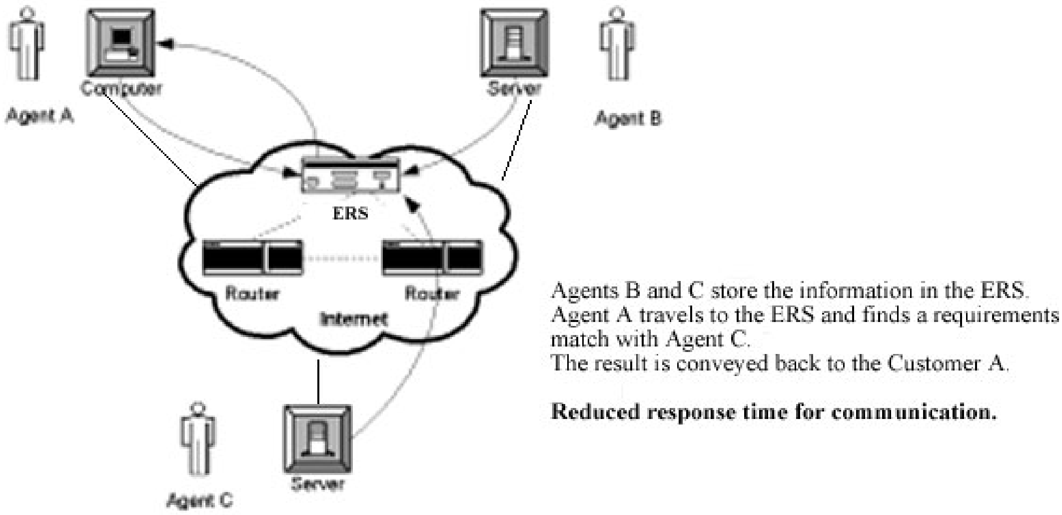

Figure 2: Inter-agent communication using the ERS

One problem that we anticipate is the memory requirement of the ERS. As the popularity of the e-Marketplaces increases, the number of merchants' agents that are registered with the router in the ERS to store the information of their product or services may rise exponentially. We propose to handle this situation by introducing timestamps. Each product or service must be associated with a timestamp and if the time limit is exceeded, details of the product is considered to be obsolete and will be removed from the memory together with that of its agent. Thus it is the duty of the owner of the agent to re-send an updated agent. This is similar to the timestamp method used in web caching (Dingle, 1996). Even when the timestamp is not expired, a merchant's mobile agent may invoke a re-validation from the merchant's origin server as to the accuracy of the data that it contains. After visiting all the ERS systems in its list, if a customer's agent does not find a suitable match, it can still function as a normal mobile agent and traverse the Internet looking for a suitable match.

The advantage of this method is that we terminate the customer's mobile agent's path at the server of the ERS and the packets of the agents will not travel end-to-end across the whole Internet to communicate with the merchants' or brokers' agents. The merchants' agents are first stored in their registered ERSs. Depending on the priority level of the service contract, these agents are multicast to the other ERSs that are owned by different ISPs. These reasons contribute to the reduction of network traffic dramatically as well as to decrease the response time. This simple and efficient method of inter-agent communication is therefore a valuable solution considering the growing popularity of agent technology and e-Commerce.

ALLOCATION OF AGENTS TO AN ERS

The solution of ERS may not be scalable unless we have a mechanism of assigning the merchants' agents to a particular ERS or a set of ERSs. This can be done in the following way. The agents can be grouped according to the tasks they perform and the agents belonging to the same task group can be assigned to a particular set of ERSs. A second level of differentiation between the agents is done according to the type of service or product that they want to sell. A further subdivision according to further details of the product or service is also possible. A hierarchy of this fashion limits the number of agents in a group to a manageable size and the required memory in the ERS will not be a problem. In this design, a trade off between the memory size, processing power, the number of levels of the hierarchy and the number of agents in the ERS has to be made. To implement this practically, each attribute in a certain level of the hierarchy has to be represented by a number. Thus the ERS that an agent belongs to is represented by a unique string of numbers. After arriving at the registered ERS, the agent is directed to the ERS that has the product or service specified in the agent. As all the products or services that have very similar attributes find themselves together in an ERS, this method increases the probability of finding a match as well. However, each ERS has to maintain the mapping of the number string of the product or service to a small set of IP addresses or names of the other ERS servers in its database. For example, if Customer A dispatched a mobile agent to the registered ERS to find a medication, the mobile agent will receive the IP addresses of all the ERS systems that fall within its service contract and that store details of medications of merchants. The number string in this case may have only two levels, i.e. it is a product and the product type is medication. In contrast to the TabiCan, there are no market advertisers in this approach and the customers' agents are directed systematically to other e-Marketplaces.

CONCLUSION

With the increased demand for e-Commerce transactions in the current Internet-centred marketplace, existing resources will soon be exhausted due to the nature and frequency of these transactions. In B2B and B2C transactions the mobile agents have to travel all the way through the Internet from the source to the destination. Besides, even after travelling to the destination and after processing, the mobile agent may still not find a suitable match to execute a successful transaction. In that event, the mobile agent has to re-traverse the Internet until it meets a matching merchant's agent. This contributes to an increase in the Internet traffic, congestion, response and decision times and eventually, the cost involved. An emerging approach that was implemented in TabiCan provides an e-Marketplace at the owner's site. This may lead to the creation of a large number of e-Marketplaces by many individuals or organisations that specialise in a few products or services. We believe that this solution can also lead to a heavy increase in the Internet traffic and also will not reduce response time significantly.

We have introduced a new framework for inter-agent communication that takes place in an ERS, creating a virtual e-Market place in it. Negotiations in the ERSs can be performed using the existing mobile agent applications. In contrast to the inter-agent communication methods that are executed at the merchant's site or at the location of the owner of e-Marketplace, this approach should reduce decision and response times, network congestion and the cost. The approach is to trade off memory, processing time, number of levels of the hierarchy and the number of agents for the network capacity. The physical location of the ERS can be associated with the core routers of the ISPs of the customer and the merchants. This has the advantage of providing agent transfer between the routers of the ERS systems, using a method similar to the inter-cache communication in web caching field, which is much mature than the agent communication field. As future work, we plan to develop a prototype system incorporating an ERS to investigate the performance.

REFERENCES

Breugst, M. and Magedanz, T., (1998), Mobile Agents - Enabling Technology for Active Intelligent Network Implementation, *IEEE Network*, 12(3), 53 –60.

CDR, (1998), *Java Agent Template Lite,* Center for Design Research, Department of Mechanical Engineering, Stanford University, Stanford CA, URL: http://java.stanford.edu/, Accessed 11 May 2002.

Chih-Lin, H. and Chen, W.S.E., (2000), A Mobile Agent-based Active Network Architecture, *Proceedings. Seventh International Conference on Parallel and Distributed Systems*, 445 –452.

Dasgupta, P., Narasimhan, N., Moser, L.E, Melliar-Smith, P.M., (1999), MAgNET: Mobile Agents for Networked Electronic Trading, *IEEE Transactions on Knowledge and Data Engineering*, 11(4), 509 –525.

de Silva, R. G., Dewar, W.J. and Low, G.C., (2001), Inter-agent Communication in an Active Networked E-Marketplace - 2nd International WE-b Conference We-b 2001, Perth, Nov 2001, 248-254.

de Silva, R.G., Li, H. and Chu , K. H., (2002), An Inter-agent Communication Protocol for Active Routers, *IEEE Asia-Pacific Conference on Circuits and Systems*, October 2002, Bali, Indonesia (accepted paper).

Dingle, A., (1996), Cache Consistency in the HTTP 1.1 Proposed Standard, http://w3cache.icm.edu. pl/workshop/talk4/, Accessed 2 May 2002.

Feldman, S., (2000), Electronic Marketplaces, *IEEE Internet Computing*, 4(4), 93 –95.

Finin, T., Labrou, Y., Yun Peng, (1998), Mobile agents can benefit from standards efforts on interagent communication, *IEEE Communications Magazine* , 36 (7) , July 1998, 50 –56.

Lange, D. B and Oshima, M., (1998) *Programming and Deploying JAVA Mobile Agents with Aglets*, Addison Wesley Longman, Inc., Massachusetts, NY.

Tokumaru, H., Kinoshita, K., Yamai, N., Murakami, K., (2001), A fast and efficient dynamic routing for inter-agent communications, *Proceedings of the International Conferences on Info-tech and Info-net,* ICII 2001,vol 4, 2001, Beijing, 73 –78.

COPYRIGHT