

Research on Mobile Agent-based E-Commerce System Framework

Wenna Liu¹, Deli Yang²

School of Management, Dalian University of Technology, Dalian 116024, China

¹ wenal@sina.com, ² somdyang@dlut.edu.cn

ABSTRACT

Electronic commerce is one of the most important application areas of mobile agent technology. A secure mobile agent system (SMAS) model is established for e-commerce environment. The SMAS consists of mobile agents and mobile agent server. A security shell module in the mobile agent architecture is responsible for mobile agent protection. Surveillance agent running in the agent server provides the host against attacks from malicious agents. Based on the SMAS, mobile agent-based e-commerce system framework is built up and realized in the end.

Keywords: electronic commerce, SMAS, empirical study, mobile agent

1. INTRODUCTON

Web-based e-commerce have developed through several phases: browsing static web pages, using mutual web tables, processing B2C commerce, till developing B2B application integration. More than half of the enterprises still rest on the phrases of static web pages and simple e-commerce. Up to present, the main form of e-commerce is B2C, which exhibiting its information through HTML, and many of its processes is launched by manual work. With the emergence of B2B and other more complicated commerce pattern, tradition e-commerce technique pattern could not guarantee for validity and could not complete tasks in time. E-commerce of next generation demands for automatic Web dealing initiated by application, decrease of error easily arisen in manual process, and efficiency. Automatization, intelligentization and integration should be embodied by the e-commerce technique pattern in next generation. More research in collaborative information technique and distributive computing pattern in new generation is the main approach to solve the problem. It will promote the development of collaborative e-commerce system that is the third generation of e-commerce. Due to its connatural advantage, mobile agent is applied in next generation e-commerce as a new technology.

Therefore, the primary aim of this paper is to give an architectural model and a framework for implementing mobile agent-based e-commerce system. Intelligent and mobile agents are applied to current e-commerce technology to meet autonomy, intelligence and integrity requirements. For the special requirement of electronic commerce, system security is the one of the most important elements to a successful system. The original idea is to establish a secure mobile agent system (SMAS) firstly, and then to build up the mobile agent-based e-commerce system framework based on the SMAS to ensure transmission security.

The remainder of this paper has the following structure: Section 2 gives an introduction to mobile agent firstly, and then establishes a secure mobile agent system

model for e-commerce. In the last portion of this section, a mobile agent-based e-commerce system framework is brought forth. A logic verification of the system framework is demonstrated in section 3. Section 4 describes some issues relevant to mobile agent applied in e-commerce. And the section 5 gives a conclusion.

2. MOBILE AGENT-BASED E-COMMERCE

2.1 Mobile Agent Introduction

The Object Management Group defines a software agent as "a computer program that acts autonomously on behalf of a person or organization" [1]. The following properties characterize agents:

- pro-active (support of the user's work)
- adaptive (learning the user's preferences or the ability to work on different platforms)
- autonomous (limited communication with its creator)
- intelligent (making 'intelligent' decisions)
- mobile (can actively migrate in networks to different systems and move directly to the local resources, like databases or application servers)

Mobile agents are programs that can migrate from host to host in a network, at times and to places of their own choosing. The state of the running program is saved, transported to the new host, and restored, allowing the program to continue where it left off. Lange and Oshima [2] give reasons why to use agents: e.g. improvements in latency and bandwidth of client-server applications and reducing vulnerability to network disconnection.

2.2 Mobile Agents in E-Commerce

Mobile agents are well suited for electronic commerce. A commercial transaction may require real-time access to remote resources such as stock quotes and perhaps even agent-to-agent negotiation. Different agents will have different goals, and will implement and exercise different strategies to accomplish these goals. We envision agents that embody the intentions of their

creators, and act and negotiate on their behalf. Mobile agent technology is a very appealing solution to this kind of problem.

An electronic commerce transaction may be viewed in terms of four different phases^[3]: product brokering, merchant brokering, negotiation, payment and delivery.

Product brokering consists in the gathering of information about the product that is going to be bought.

Merchant brokering involves the evaluation of a set of alternatives in order to make the purchase. Making the decision implies considering all the tradeoffs that the various products offer: price, warranties, delivery time, and others.

During the negotiation phase the agent settles the final terms of the commercial transaction. The characteristics of the market directly influence the outline of this phase. In markets where prices and characteristics are fixed, negotiation may not even exist.

Finally, in the purchase and delivery phase of the transaction, the agent actually makes the acquisition and delivers the money (or its electronic equivalent) against the goods.

2.2.1 Secure Mobile Agent System

A number of advantages of using mobile code and mobile agent computing paradigms have been proposed^[4, 5, 6]. One of the main obstacles to the widespread adoption of mobile agents is the legitimate security concern of system developers, network administrators, and information officers. It has been argued that once the security issues have been resolved and a collection of security mechanisms have been developed to counter the associated risks, then the users of mobile agent technology will be free to develop useful and innovative solutions to existing problems and find a wide array of application areas that will benefit from this technology. Using this collection of security mechanisms to mitigate agent-to-agent, agent-to-platform, and platform-to-agent security risks may, however, introduce performance constraints that could dictate design decisions or negate the benefit of using mobile agents for certain applications. Considering the special security requirements in e-commerce, a secure mobile agent system (SMAS) framework is presented in this paper.

SMAS consists of mobile agents and agent server (e.g. agent facilitator). Mobile agent has six modules: security shell, environment interaction module, state denotation module, task execution module, routing policy module and log file. The architecture of mobile agent is shown in figure 1.

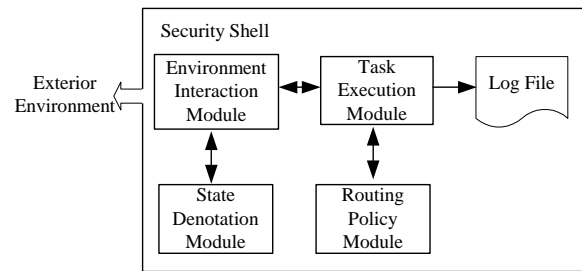


Figure 1 Mobile Agent Architecture Model

Security shell is responsible for protecting the agent from attacks of malicious hosts and other agents. The main functions include encrypting and decrypting the whole agent, authorizing the visit host, deploying the different security policy according to the different trusted level of host.

Environment interaction module is responsible for the maintenance of communication between agent and exterior environment that includes sensor (apperceive outside environment), message processor (process inter-messages, data communication between agent and environment based on KQML) and reactor (i.e. effector, output the result to outside environment).

Task execution module is the execution module of agent, which consists of action execution module and result integration module. According to the security strategy to different methods and data deployed by security shell, task execution module executes corresponding method, and makes integrated analysis of results for its goal.

State denotation module is responsible for recording the state of agent, which includes attribute value of agent and the sequence of agent's running environment. This makes agent to run successfully when it is hold up or resumed. Agent's state can also be recorded in fixed interval for its resuming in abnormal circumstance.

Routing policy module plans agent's migration route. There are two feasible routing policies, one is fixed routing, and the other is dynamic routing based on formula and catalog services.

Log file records every sensitive instruction and manipulation on agent executed by agent platform, it's used for audit afterwards. Log file must be encrypted so that it couldn't be juggled.

Mobile agent server (mobile agent service facilitator) is based on agent transfer protocol (ATP), it provides essential EE(execution environment) for agent's migration, execution, and other functions and services such as dispatch, reception, recovery, security management and service transfer. Mobile agent server includes security management, agent executing environment, agent API, agent service environment, communicating API, as shown in figure 2.

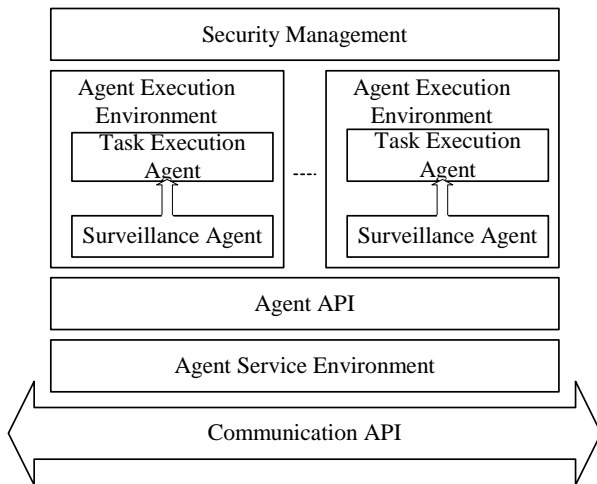


Figure 2 Mobile Agent Server Structure

Security management is responsible for the security of agent server and the agent executing on it, it takes authentication, encryption, visit controlling and other measures to protect system security.

Every agent server can process several agent EEs. Agent EE is the resource distributed for every agent by the agent server. It runs agent under the control of security management module. Every EE has two kinds of agents: task execution agent and surveillance agent. Task execution agent runs the program on behalf of agent user. Surveillance agent keeps an eye on the execution situation of the task agent. It can terminate the task agent once there is any unauthorized or illegal operation.

Agent API is the unique interface between agent and agent server. It provides many characteristic operations of mobile agent that can be utilized to finish the distributed task for the agent. For instance, agent calls Jump procedure explicitly before migrating to another host. Jump procedure captures the whole state image of the mobile agent and transfers the agent to the destination host. On the destination host, agent server loads the agent to the relevant agent EE and recovers the agent from the hold point. The primary agent will be terminated on the local host after the agent's state image being captured, transferred and recovered successfully on the destination host.

Agent service environment manages lifecycle, event and persistence for agent. Lifecycle management realizes creation, dispatch, migration, reception, storage and termination for mobile agent, event service provides agent communication mechanism for collaboration with other agents or application system, which supplies essential condition for agent collaboration. Persistence service ensures agent execution's persistence through corresponding mechanism, it helps agent to restart accurately when there is system or network disaster before agent reach its destination.

Communication API is responsible for transmitting mobile agents, data and messages, including network protocol API and email API.

2.2.2 Mobile Agent-based E-Commerce

Based on above construction of mobile agent system, this paper brings forth the mobile agent-based e-commerce system framework. Its model is shown in figure 3.

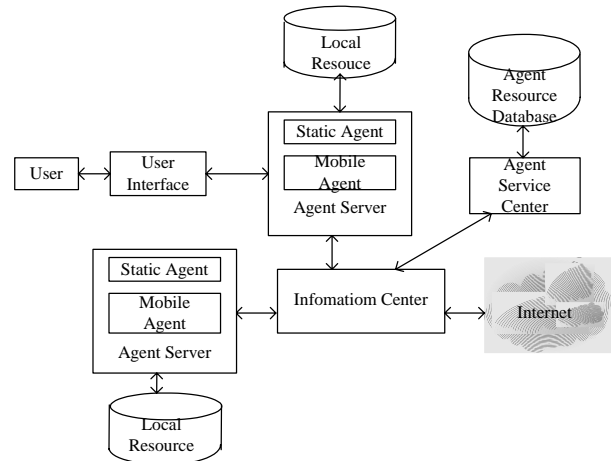


Figure 3 Mobile Agent-Based E-Commerce System Framework

In this system, agent service center provides catalog service, registers mobile agent's basic information, maintains agent's lifecycle, cancels illegal agent in time and provides agent query service. In order to enforce system security, especially for mobile agent protection, agent service center also maintains a table which records the history of host visitation, it gets task accomplish information of agent in visitation from its log files, calculates the host's credit degree, puts the result in the history table and re-calculates it so that mobile agent could optimizes routing strategy. Service center need to provide host service query function in order to help agent affirm the servers on which its task could be accomplished, and choose the best route.

Information center is responsible for security control of the whole E-Commerce system, it generates certain inbreak inspection agents, sends them to each sub-system, monitors real time condition of system execution, collects relative information of inbreak event, analyzes inbreak event, traces inbreak route and confirms its beginning. Control center takes corresponding measure when it received inbreak report and analyzed it. Control center is also responsible for downloading patches of application and operation system, upgrading virus-preventing software, after this it dispatches mobile agent to install them on every node in the system.

Following is the workflow of system:

- (1) User submits a request through user interface.
- (2) Agent Server analyzes the request and creates static agent or mobile agent according to the request of task, then registers the agent's information in agent service

center.

(3) Static agent visits local resource under security control of agent server, completes user's request and returns the result to user. In the meantime, it registers agent's logout information on agent service center.

(4) Mobile agent registers its information on agent service center, queries for host service table, returns agent server, works out migration route or plan according to routing strategy.

(5) Mobile agent queries the history table of host visiting on agent service center, gets credit degree of host on the route, returns the optimized migration route of agent server.

(6) Mobile agent sends out migration request to server, agent server makes agent hung up, encrypted, packed and sends it to destination host.

(7) Destination host accepts agent, authenticates each other.

(8) Destination host carries out security strategy according to authentication result, assigns limited execution environment for agent, and generates monitor agent to monitor the execution information of mobile agent.

(9) Mobile agent carries out different security strategy according to authentication result, executes task under assigned resource. If necessary, it could make alliance with other agents and accomplish the task collaboratively.

(10) Agent goes through hosts in routing table and completes its task, collects the results, returns it to user and submits log files to agent service center, registers logout information.

3. VERIFICATION

At present, many technology research and system development based on agent have been gotten along at home and abroad. However there is little research on agent system verification. Yanxiang He and Xinmeng Chen introduced an agent-mediated system verification method in 2001^[11]. The definition of agent-mediated system verification can be described as follow: a system contains one or more claimed agent(s), it will be changed when some other claimed agent(s) add(s) to the system. That means:

(a) System behavior transformation doesn't indicate other process's running slows down the system processing speed. Otherwise it shows the claimed agents apperceive each other to a degree, and they adjust their actions accordingly.

(b) New added agent must have the same style with the claimed agents, viz., although it has different goal, knowledge and belief, it should have the same architecture and function.

(c) The existing agents and the new added agent don't always communicate with each other, and they not always be autonomous, persistent or intelligent.

According to the definition, following test should be done. We suppose X have two instances. If the behavior

when there is another instance to exist of each instance and its behavior while there are no other instances to exist are different, that indicates X is an "agent". On the contrary X is not an "agent". The verification has nothing to do with agent's mobility.

For example, a user submits a purchase requirement, two instances of information collection agent, agent1 and agent2, are created. Agent1 and agent2 move in the network according to their migration route and collect the required commodity information. Each agent apperceive environment automatically and make action according to the environment. When one agent's reaction reaches the existence of another agent, they communicate with each other and get the new task result and execution decision by negotiation. Then they regulate their behaviors based on the new strategy. So they are "agent" on the agent-mediated system verification definition mentioned above. The rest may be deduced by analogy, we can prove that the system established on the framework introduced in section 2 is an agent-based system.

This verification method is based on agent's sociability. For security and performance verification of the system, demonstration and simulation verification are needed.

4. RELATED AND FUTURE WORK

Security insurance and system reliability are the foundation and key for the popular application of the mobile agent-based e-commerce. In this paper, secure mobile agent architecture is designed for protecting agent from attacks of malicious host and other agents; surveillance agents are added for protecting the host from attacks of malicious agents; XML-PKI technology is provided for encrypting the sensitive data, code and states for transaction security of agent. Disaster recovery technology is applied to buildup system reliability.

There are many promising new research on mobile agent technology and its applications in recent years^[8,9,10]. However we should see that we must solve some more technology obstacles on our way. The first one is the issue of agent standardization. FIPA and MASIF are working on that and there should be some standards about mobile agent system framework and so on. For agent in e-commerce, it is still in its infancy. The main trend in this area is to attack four problems: negotiation, information discovery, ontology and security. Furthermore, the use of electronic money in conjunction with mobile agent is particularly challenging. The integration of digital payment system into mobile agent-based e-commerce is important to realize the potential e-market. In a word, there are many jobs should be done for the mobile agent-based e-commerce application in the future.

5. CONCLUSION

Mobile agent-based e-commerce system framework mentioned above utilizes the virtues of mobile agent technology fully. It can realize the electronic transaction automatically and intelligently. It also can provide the better security protection for effective and convenience business cooperation and collaborative information sharing between enterprises. To realize the system concretely, Java language is adopted to implement the mobile agent for platform independence. Mobile agent communicates in KQML (Knowledge Query and Manipulation Language) with each other and accesses services provided by agent server. Mobile agent server adopts ATP (Agent Transfer Protocol) to dispatch and receive mobile agents.

ACKNOWLEDGEMENT

The research work has got a lot of people's help and energetical support. Especially thank professor Yang for his guidance and support.

REFERENCES

- [1] Crystaliz, Inc., General Magic, Inc., GMD Focus, and IBM Corp, "Mobile Agent Facility Specification Technical report", *OMG*, 1997.
- [2] Danny B. Lange, Mitsuru Oshima, "Seven Good Reasons for Mobile Agents", *Communications of the ACM*, Vol 42, No. 3, pp88-89, March 1999.
- [3] P. Maes, R. Guttman, and A. Moukas, "Agents that Buy and Sell", *Communications of the ACM*, vol. 42, 1999.
- [4] Danny Lange, Mitsuru Oshima, "Programming and Deploying Java Mobile Agents with Aglets", *Addison-Wesley*, 1998.
- [5] Jonathan Smith, "A Survey of Process Migration Mechanisms," *Operating Systems Review, ACM Special Interest Group on Operating Systems*, Vol 22, No. 3, pp28-40, July 1988.
- [6] David Chess, Colin Harrison, and Aaron Kershenbaum, "Mobile Agents: Are They a Good Idea?", *IBM Research Report RC 19887 (88465)*, December 1994.
- [7] Christian F. Tschudin, "Open resource allocation for mobile code", *In Proceedings of The First Workshop on Mobile Agents*, pp186-197, April 1997.
- [8] Birk A, "Boosting cooperation by evolving trust", *Applied Artificial Intelligence*, Vol 14, No. 8, pp769-784, 2000.
- [9] Castelfranchi C, Falcone R, "Trust and control: dialectic link", *Applied Artificial Intelligence*, Vol 14, No. 8, pp799-823, 2000.
- [10] Zacharia G, Moukas A, and Maes P., "Collaborative reputation mechanisms for electronic marketplaces", *Decision Support Systems*, Vol 29, pp371-388, 2000.
- [11] Yanxiang He, Xinmeng Chen, "Design and application of agent and multi-agent system", *Wuhan University Press*, pp203-235, 2001.