Mobile Agents for Secure Electronic Transactions

P. KOTZANIKOLAOU, G. KATSIRELOS, V. CHRISSIKOPOULOS
Department of Informatics
University of Piraeus
80, Karaoli & Dimitriou, Piraeus 185 34
GREECE

Abstract: - In the area of electronic commerce the technology of mobile trade agents can be used in market research, buyer-merchant negotiation and on-line auctions. Although the benefits resulting from the use of such intelligent assistants for the end-users are not argued, it is empirically confirmed that Internet buyers and merchants will use them widely, only when convinced that mobile trade agents are secure. This paper presents an agent-oriented model for collecting and evaluating purchase contracts, signed by Internet merchants. It aims to confront the security risks derived from mobile trade agents. The model uses a master - slave distributed agent architecture and proposes the authentication of mobile agents to shopping servers, through agent permission-tokens. *CSCC'99* Proc.pp..4001-4006

Key Words: - Electronic commerce, mobile trade agent, master-slave agent model, agent permission-token

1. Introduction

As the number of web users is rapidly increasing, more and more enterprises tend to have some sort of on-line presence. Others convey part of, or their entire commercial activities on the Internet. The reason behind this vast growth is that e-commerce offers several advantages over traditional ways of doing business. Organizations realize the competitive advantages they gain, going on-line. Customers confront e-commerce as an alternative, complementary way of purchasing goods.

The chaotic structure of the Internet makes it difficult for potential Internet buyers to manually visit many virtual enterprises and compare a sufficient number of competitive offers. One approach to this problem is special intermediate trade services that use distributed object technology, namely electronic brokers (e-brokers). They provide services such as searching for a suitable business partner or product, negotiating the terms of a deal and ensuring delivery of goods. An example of an e-broker, which uses OMG's CORBA as a distribution infrastructure, is presented in [6]. Other examples of such intermediaries can be found in [10, 11].

A different approach to the problem is the use of trade agents. Trade agents are autonomous software entities that can use artificial intelligence and behave in a smart way, offering new paradigms for electronic trading. They interact with other agents or people and act as users' representatives. They can browse for products using smart searching techniques, gather shopping information, analyze collected data from hosts, negotiate on behalf of their user/owner, and present a (sub)set of the negotiation result to the user. Mobile trade agents are able to migrate across execution environments, roam into a network and return into their initial source. They can work in a distributed way i.e. different parts of the agent can be running on different hosts. Mobile trade agent systems offer extended capabilities, because mobility and autonomy make permanent connections unnecessary [3]. They also provide low-bandwidth connections, asynchronous interaction and better support for heterogeneous environments [4]. However, security is the main concern for mobile trade agent systems [1, 2, 3, 7, 8]. Merwe and Solms [1] proposed a secure intelligent trade agent system, while Yi et al. [2] present a trade agent system that relies its security on an Agent Service Center (ASC) which sends roaming agents to a number of electronic shops after buyers' request. Zapf et al. [3] specify security requirements for mobile agents in electronic markets and implement some of them in their agent system AMETAS.

In this paper, we propose a secure distributed agent model for collecting and evaluating purchase contracts, which uses a collection of n+1 collaborative agents in order to negotiate with n shops. One of the

agents, named master agent, is static (it never leaves user's computer), while the rest of the agents, denoted as slave agents, are mobile. The master agent is responsible to provide each slave agent with a permission-token for a discrete shopping server respectively. Each slave agent migrates to a shopping server, negotiates for specific products and returns to its initial source with a purchase contract, signed by the electronic shop. The master agent evaluates the signed contracts and presents the results to the buyer, who can purchase the required goods from the shop that signed the optimal contract.

The organization of this paper is as follows: In section 2 we present the proposed trade agent model. In section 3 we discuss the security that this model provides both to the buyer and to the merchant side, while in section 4 we discuss the advantages of this approach.

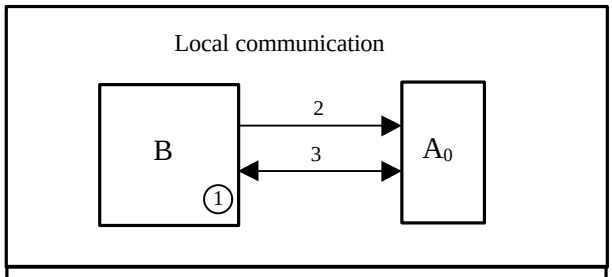
2. The Proposed Model

We consider a buyer B who wants to purchase some products from virtual enterprises in the Internet. The buyer can visit *n* electronic shops S_1, S_2, …S_n and compare *n* competitive offers. Instead of manually surfing through the web sites, he assigns the process to a group of collaborative agents A_0, A_1, … A_n. Agent A_0 is the static, master agent while the remaining ones are the mobile slave agents. We divide our analysis in five phases. In the first phase, the buyer sets up the master agent and interacts with it, in order to produce the shopping requirements for a specific purchase. In the second phase the master agent communicates with the electronic shops S_i $i\in[1,n]$, to ask for negotiation permission-tokens for the slave agents A_i $i\in[1,n]$. In the next phase, the master agent generates the slave agents and provides each of them with a permission-token, to be used as authentication proof for an electronic shop. The slave agents migrate to the servers of the electronic shops and negotiate on behalf of the buyer. In the fourth phase, the slave agents return to their source with purchase contracts, signed by the electronic shops. The master agent is responsible for the evaluation of the contracts. Finally, the buyer uses a payment system to purchase the required goods from the electronic shop that proposed the optimal offer, under the terms of the specific signed contract.

For simplicity, we denote with P_X, S_X the public and secret keys of the entity X, where $X\in\{B, A_0, S_i\}$, created with a public key infrastructure (PKI), (e.g. RSA). Cert(X) denotes the digital certificate of the entity X, $K_X[M]$ the encryption of a message M using the key K_X and $h(\ )$ denotes a hash function (e.g. MD5).

2.1 Setting up the master agent

The buyer B initializes the master agent A_0 and starts the process of requisitioning competitive purchase contracts. The buyer obtains a public and secret key for the master agent, using the PKI (Fig.1).



1. B: initialize A_0, P_{A0} , S_{A0}
2. B→A_0: **(**Cert(B) | S_{A0} | P_{A0} | ID(A_0) | $S_B[h(P_{A0}, ID(A_0))]$ **)**
3. B↔A_0: SR

Fig.1. Buyer - master agent interaction

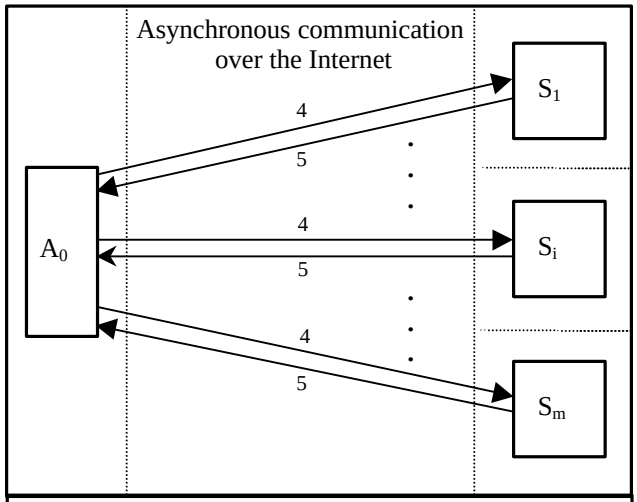
He uses a pseudorandom generator to create a unique identifier ID(A_0) for the master agent and he authenticates the master agent as his legal representative, by providing it with his digital certificate Cert(B) and signing its identification number and public key. These steps can be repeated if the buyer suspects violation of the master agent's integrity. Then the buyer and the master agent exchange messages interactively, to reach in the specific shopping requirements (SR) of the buyer. The master agent does more than waiting for answers in predefined questions; i.e. it is able to learn from the buyer's previous behavior, guide him and/or make suggestions. At the end of this process, the master agent knows the buyer's shopping requirements SR and is able to continue the request, without bothering the buyer again, until the purchase phase.

2.2 Issuing of the agent permission-tokens

The master agent communicates with the electronic shops S_i $i\in[1,m]$, to issue permission-tokens for the slave agents. The buyer initializes a list with the URLs of his favorite virtual enterprises. The master agent consults this list to contact with the particular electronic shops and maintains it by having access rights to append more addresses to the list or delete others, based on the buyer's preferences. Namely, the

master agent learns from the buyer's previous shopping behavior his favorite virtual enterprises.

The master agent contacts a shopping server S_i and requests a permission-token for future interaction of its slave agent A_i with the shop, by sending message (1), (Fig.2). R_i is a unique request number, generated by the master agent.



4. $A_0 \rightarrow S_i$: $\mathbf{P_{Si}[}$ Cert(B) $|P_{A0} | ID(A_0) | R_i |$ $S_B[h(P_{A0},ID(A_0))]$ $\mathbf{]}$ **(1)**

5. $S_i \rightarrow A_0$: $\mathbf{P_{A0}[}$Cert(S_i)$|$ $(ID(A_0),P_i,R_i,T,T_{exp})|$ $S_{Si}[h(ID(A_0),P_i,R_i,T,T_{exp})]$ $\mathbf{]}$ **(2)**

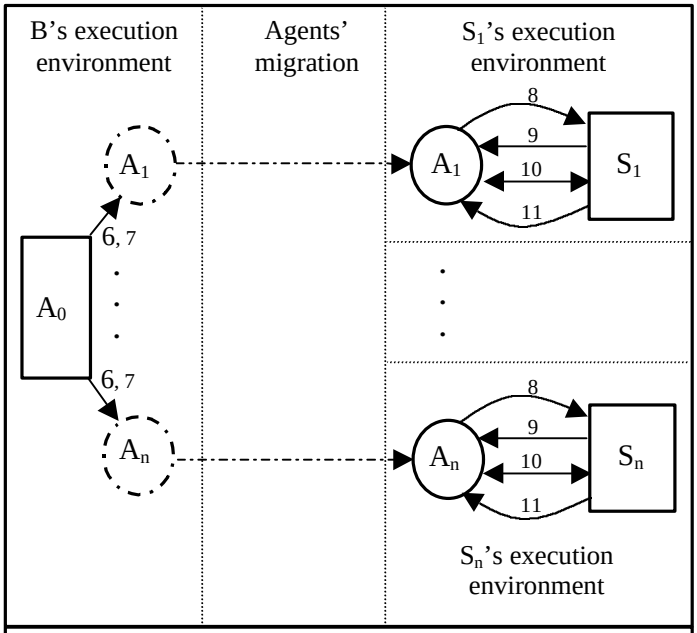
Fig.2. Interaction between master agent and the electronic shops

The shop S_i receives message (1), decrypts it and checks the validity of Cert(B) and the buyer's signature: $P_B[S_B[h(P_{A0},ID(A_0))]]$ $h(P_{A0},ID(A_0))$. If the validation succeeds then the shop S_i accepts the master agent as the buyer's representative and sends to it an agent permission-token $(ID(A_0),P_i,R_i,T,T_{exp})$ in message (2). P_i is a unique permission number generated by S_i, while T, T_{exp} are the current and the expiration time of the permission-token, respectively. The master agent receives message (2), decrypts it and checks Cert(S_i) and the validity of the signature of the electronic shop on the permission-token: $P_{Si}[S_{Si}[h(ID(A_0),P_i,R_i,T,T_{exp})]]$ $(ID(A_0),P_i,R_i,T,T_{exp})$. If all the exchanged messages are valid, the master agent accepts the authenticity of the permission-token.

The master agent repeats the process and collects *m* permission-tokens. When the slave agents have used some of them and the number of the remaining permission-tokens reaches a predefined number *l*, the master agent automatically restarts this phase. Although the buyer must be on-line to enable the master agent to perform these steps and thus the mobility is compromised, the communication can be asynchronous. In order to minimize the added burden, the master agent can run as a background process and collect permission-tokens from electronic shops, while the buyer is on-line for irrelevant purposes.

2.3 Negotiation between the slave agent A_i and the electronic shop S_i

The master agent starts the negotiation with the electronic shops by generating *n* slave agents A_i $i \in [1, n]$ and providing each of them with a permission-token for the server S_i, as shown in Fig.3. The number *n* of electronic shops that the master agent will ask for permission-tokens vary depending on the buyer's special needs; that is, from small values for a quick decision, to large values for an optimal decision.



6. $A_0 \rightarrow Ai$: $\mathbf{P_{Si}[}$ Cert(B) $|$ $(ID(A_0),P_i,R_i,T,T_{exp})$ $|$ $S_{Si}[h(ID(A_0),P_i,R_i,T,T_{exp},)]$ $\mathbf{]}$ **(3)**
7. $A_0 \rightarrow A_i$: Cert(S_i), SR, P_{A0}
8. $A_i \rightarrow S_i$: **(3)**
9. $S_i \rightarrow A_i$: Cert(S_i)
10. $A_i \leftrightarrow S_i$: SR
11. $S_i \rightarrow A_i$: $\mathbf{P_{A0}[}$ SC_i, S_{Si} $[h(SC_i)]$ $\mathbf{]}$

Fig.3. Slave agents' – Servers' interaction

Also, the master agent provides each slave agent A_i with the buyer's shopping requirements SR. Now A_i is ready to migrate to the shopping server S_i and negotiate with it based on the buyer's shopping requirements SR.

For security reasons, the shop S_i requires authentication proof from the slave agent A_i before accepting its execution request. The slave agent A_i

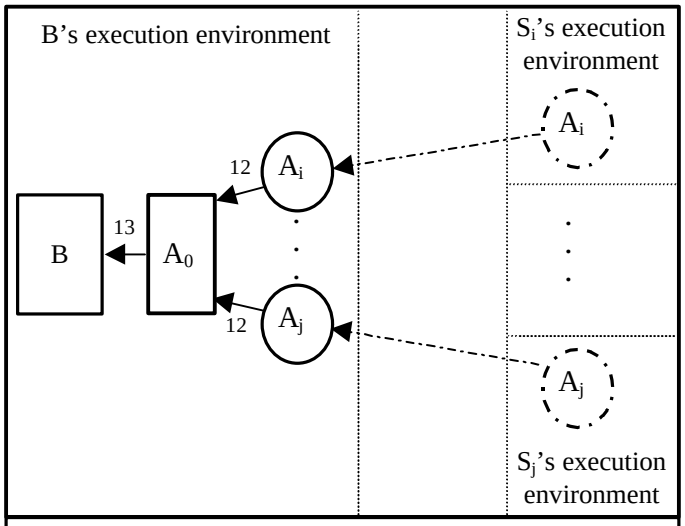
gives to the shop S_i such proof by sending message (3). The electronic shop S_i receives message (3) and checks the validity of Cert(B) and $S_{Si}[h(ID(A_0),P_i,R_i,T,T_{exp})]$. Also it checks the expiration time T_{exp} of the permission-token.

If the verification process succeeds, the electronic shop S_i provides the slave agent A_i with execution environment. The slave agent asks for the specific good(s) under the specific requirements SR. The result of the communication is a purchase contract SC_i, signed by the electronic shop S_i. The slave agent receives the encrypted contract and terminates the negotiation.

2.4 Evaluation of signed offers

The slave agents A_i, $1 \le i \le n$, return to their source and provide the master agent with message (4) (Fig.4). It is pointed out that we take into account the possibility that not all of the *n* initial slave agents return to their source. Malicious hosts, network failure or other reasons could cause the loss of some slave agents. Nevertheless, the evaluation of the remaining *k* ($k \le n$) results can continue.

The master agent decrypts message (4) and verifies the signed contracts. In case some of the contracts are not signed, the master agent rejects them and continues the evaluation of the remaining.



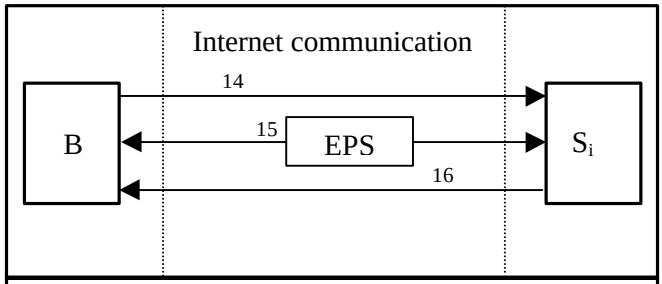
12. $A_i \rightarrow A_0$: $\mathbf{P_{A0}[}$ Cert(S_i) | SC_i | S_{Si} [$h(SC_i)$] $\mathbf{]}$ **(4)**
13. $A_0 \rightarrow$ B: $\mathbf{DS[}SC_i, S_{Si}[h(SC_i)]\mathbf{]}_{i=1}^{k}$

Fig.4. Collection and evaluation of the signed purchase contracts

The master agent inputs every valid contract to a decision support mechanism (DS) and evaluates them, based on the buyer's personal requirements. For example, if for a specific purchase the buyer demands fast delivery, the results are ranked based on the delivery period, as the most significant criterion.

2.5 Purchase phase

The buyer uses the signed purchase contract that was evaluated as the optimal in the previous phase and buys the product(s) from the particular electronic shop S_i. The electronic shop checks its signature on the contract and if it is valid it cannot deny the terms of the contract. The buyer uses an electronic payment system (EPS) that is accepted by the electronic shop and pays for the requested products. Finally, the shop delivers the goods (Fig.5).



14. $B \rightarrow S_i$: $\mathbf{P_{Si}[}$Cert(B), SC_i, $S_{Si}[h(SC_i)]$ $\mathbf{]}$
15. $B \leftrightarrow S_i$: Payment
16. $S_i \rightarrow$ B: Goods Delivery

Fig.5. Execution of the optimal contract

Obviously, if the requested goods are physical and not "digital" (i.e. software), the delivery of the goods is executed off-line.

3. Security Analysis

It has been shown [8, 9] that the security of mobile agent models can be divided into two broad areas; the security of mobile agents against dishonest servers and the security of Internet servers against malicious agents.

3.1 Security of the electronic shops' servers

In the proposed model, the servers of the electronic shops require adequate authentication proof, before accepting further interaction with an agent. The master agent A_0 is authenticated as the buyer's legal representative in the first phase, where the buyer provides the master agent with his certificate and a signature on master agent's public key and identification number, $S_B[h(P_{A0},ID(A_0))]$. Later, the master agent A_0 sends this authentication proof to the electronic shop S_i. The shop can deny communicating with a master agent, which is not authenticated by a

legitimate and traceable buyer. Each slave agent A_i is authenticated to the corresponding shop S_i, by providing the shop with the permission-token that the shop had issued for the master agent. The master agent has previously provided the slave agent A_i with this permission-token.

A server can follow an effective access control policy, by taking several countermeasures in order to prevent security attacks of a malicious mobile agent [7]. In our model, a shopping server prevents such attacks by denying access to any mobile agent which has no adequate authentication proof, namely a valid permission-token issued by the electronic shop and a valid certificate of the agent's owner. The electronic shop requires that authentication proof in order to be able to link an agent with its owner, if necessary. Additionally, the server of the shop checks the code and data of an agent for viral software before it provides the mobile agent with the requested execution environment. The electronic shop provides the agent with a limited execution environment and grants access rights only for that environment. If an agent is suspect of hostile behavior, the server can suspend its rights, even destroy the agent. The confidentiality of the transferred messages through the Internet is assured by public key encryption.

3.2 Security of the agents

The master agent is static thus a malicious host cannot violate its integrity. The main security concern of the master agent is the authenticity of the shopping servers. The electronic shop is authenticated by sending to the master agent its certificate, along with the signature of the permission-token $S_{Si}[h(ID(A_0),P_i,R_i,T,T_{exp})]$.

More security concerns are related with the protection of the slave agents, since they are mobile. Each slave agent A_i migrates from the buyer's computer to the server of the shop S_i, $i \in [1,n]$. A dishonest host may deny providing A_i with execution environment or try to violate its integrity. In our model, a server has no motive to attempt the violation of the slave agent's integrity, since the slave agents do not carry any sensitive information, which could be used for any illegal purposes, by the electronic shop. In particular, each slave agent migrates and negotiates with a single shop without carrying any former negotiation results (i.e. purchase contracts signed by other electronic shops). In addition, the slave agents do not carry any secret keys, since the purchase phase is assigned to the buyer. Therefore, the slave agent does not have any information that could tempt the server of the shop.

A malicious server could cause denial of service to an authenticated slave agent with a valid permission-token or even behave in a vandalistic way and terminate the agent. It is highly unlikely that the server of an electronic shop will act this way, since the master agent knows the identity of the server that each slave agent visits. If the master agent detects hostile behavior against a particular slave agent, it can add the corresponding electronic shop in a "black list" and cease any future transactions with this dishonest shop.

In order to ensure confidentiality of the signed purchase contract that the slave agent carries, the resulting purchase contract of each negotiation session is encrypted until the agent returns to its source.

An electronic shop cannot refuse to sell the required product(s) under the terms of the purchase contract it issued, because it has previously signed it. The unsigned purchase contracts are not accepted for evaluation by the master agent.

The proposed model can overcome denial of service of some potential hostile shopping servers without restarting the whole process. In particular, if *k* out of *n* slave agents do not return to the buyer, the system can continue the evaluation of the remaining *n-k* offers. It is obvious that the credibility of the result is conversely analogous to *k*.

4. Discussion

Some trade agent systems propose the employment of one mobile agent, which visits and negotiates with *n* electronic shops sequentially, [2]. In such systems the roaming agent carries all former k negotiation results when it visits k+1 shop, so it is necessary to protect these results. In order to be able to discover a potential hostile server, the buyer needs the collaboration of a Trusted Third Party, which generates the agent on the buyer's request, monitors the process and returns the results.

Our model demands the collaboration of one master and *n* slave agents in order to negotiate with *n* shops. In addition, the phase of issuing permission-tokens seems to humble the basic advantage of mobile agents, namely working off-line. The use of *n* instead of one mobile agent does not compromise the efficiency of our scheme in an intolerable level. Each one of the *n* slave agents has the task of negotiating with a single electronic shop and to return the result to the master agent. The process can be executed in a parallel way, a significant gain over the serial way of sending the same trade agent to servers S_1, S_2 … S_n, successively. Furthermore, the need of issuing permission-tokens through on-line communication between master agent and the electronic shops can be

preformed asynchronously. The master agent can maintain a database with granted permission-tokens, sorted on the basis of homogenous shops. Thus the additional burden on the performance, imposed by this phase, can be significantly minimized, while the need for a third party, which monitors the agent migration process, is eliminated.

Each slave agent does not carry any security sensitive information when migrating to the shopping server S_i. This fact minimizes the protection needs of the mobile agents to a posteriori detecting tampering rather than a priori prevention, which would require cryptographic techniques or hardware devices, far more costly and complicated [4, 5]. An electronic shop that acts in a hostile way to an well-authenticated slave agent can be easily detected from the master agent and be suspended from future purchases.

The proposed scheme provides better fault tolerance over trade-agent schemes, which use a single mobile agent [1, 2]. If this single agent does not return to the user because of a hostile server or a network failure, the whole process must be restarted. Our scheme can overcome the failure of *k* slave agents and continue with the remaining *n-k* mobile agents. This approach is on the interest of honest shopping servers also, as their offers are not lost due to the misbehavior of a malicious server.

This framework does not offer anonymous communication. The modification of our scheme, in order to provide anonymous agent communication is under investigation. The anonymity of the payment is not related to the anonymity of the agents, because the buyer can use an anonymous payment system to execute the transaction.

Mobile agent development platforms are divided in two sets [5], weakly mobile (e.g. [12, 13]) and strongly mobile (e.g. [14, 15, 16]). We are evaluating these platforms in terms of agent mobility, remote messaging and security, in order to find the most suitable for the implementation of our model.

References:

[1] J. Merwe and S.H. Solms, Electronic Commerce with Secure Intelligent Trade Agents, *Proceedings of ICICS'97, LNCS* Vol. 1334, Springer-Verlag, November 1997, pp.452-462.

[2] Yi Xun, Wang Xiao Feng and Lam Kwok Yan, A Secure Intelligent Trade Agent System, *Proceedings of the International IFIP/GI Working Conference, TREC'98, LNCS* Vol. 1402, Springer-Verlag, 1998, pp. 218 – 228.

[3] Zapf Michael, Muller Helge and Geihs Kurt, Security Requirements for Mobile Agents in Electronic Markets, *Proceedings of the International IFIP/GI Working Conference, TREC'98, LNCS* Vol. 1402, Springer-Verlag, 1998, pp. 205 – 217.

[4] Tomas Sander and Christian F. Tschudin, Protecting Mobile Agents Against Malicious Hosts, *Mobile Agent Security, LCNS* Vol. 1419, Springer-Verlag, 1998, pp. 44-60.

[5] Giovanni Vigna, Cryptographic Traces for Mobile Agents, *Mobile Agent Security, LNCS* Vol. 1419, Springer-Verlag, 1998, pp. 137-153.

[6] Martin Bichler, Carrie Beam, Arie Segev, OFFER: A Broker-Centered Object Framework For Electronic Requisitioning, *Proceedings of the International IFIP/GI Working Conference, TREC'98, LNCS* Vol. 1402, Springer-Verlag, 1998, pp. 154 – 165.

[7] D. Chess, C. Harrison and A. Kershenbaum, Mobile Agents: Are They a Good Idea?, *In J. Vitek and C. Tschudin (eds) Mobile Object Systems,* Springer, 1996.

[8] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris and G. Tsudik, Itinerant Agents for Mobile Computing, *Technical Report, RC 20010,* October 1995, IBM T.J. Watson Research Center, NY.

[9] Fritz Hohl, Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts, *Mobile Agent Security, LNCS* Vol. 1419, Springer-Verlag, 1998, pp.92 – 113.

[10] Andersen Consulting Bargainfinder, February 1999, http://bf.cstar.ac.com/bf/

[11] Netbot Jango, January 1999, http://www.jango.com

[12] Sun Microsystem, Java, February 1999, http://www.javasoft.com

[13] IBM, Aglets Workbench, February 1999, http://www.trl.ibm.co.jp/aglets

[14] General Magic, Odyssey, February 1999, http://www.genmagic.com/technology/odyssey.html

[15] ObjectSpace, Voyager, February 1999, http://www.objectspace.com/developers/voyager/index.html

[16] IKV++, Grasshopper, February 1999, http://www.ikv.de/products/grasshopper/grasshopper.html