The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0968-5227.htm

An evaluation of a secure and trustworthy mobile agent-based e-marketplace system

Ahmed Patel
Centre of Software Technology and Management (SOFTEM), School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, KM Bangi, Malaysia and School of Computing and Information Systems, Faculty of Science, Engineering and Computing, Kingston University, Kingston upon Thames, UK, and

Wei Qi and Mona Taghavi
Centre of Software Technology and Management (SOFTEM), School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, KM Bangi, Malaysia

Received March 2011
Reviewed June 2011, August 2011, September 2011
Accepted September 2011

Abstract

Purpose – Mobile agent-based e-marketplaces are business applications that have been developed as flexible and efficient approaches to help companies or corporations to extend their businesses to outreach larger markets without regional and continental boundaries. Every distributed system is unable to avoid security problems due to the open internet environment. Mobile agent-based e-marketplaces are no exceptions. The purpose of this paper is to design and implement a framework and system of a secure and trustworthy mobile agent-based e-marketplace to overcome this problem.

Design/methodology/approach – The authors present an analysis and evaluation of a secure and trustworthy mobile agent-based e-marketplace, which was specified and prototyped. The experimental results of the implemented system are used to address the evaluation of the system. The discussion of the solution is also presented.

Findings – The evaluation and performance results show that the proposed framework and system have the ability to provide a secure and efficient e-marketplace environment for trading products. The authors draw conclusions and highlight future work on this specific research area.

Originality/value – The performance and scalability are the two most important issues for mobile-agent based systems together with their feasibility. The evaluation and performance results are used to reflect the results of the research in its entirety.

Keywords Electronic commerce, Data security, Protocols, Evaluation and performance results, Mobile agents, Digital forensics

Paper type Research paper

The authors wish to acknowledge and thank members of the research teams of the Exploratory Research Grant Scheme (ERGS) number ERGS/1/2011/STG/UKM/01/16 and the Long Term Fundamental Research Grant Scheme (LRGS) number LRGS/TD/2011/UKM/ICT/02/01 for this work. Both research schemes are supported by the Ministry of Higher Education (MOHE) under the Malaysian R&D National Funding Agency Programme.

Information Management & Computer Security
Vol. 20 No. 2, 2012
pp. 123-146

0968-5227
DOI 10.1108/09685221211235634

1. Introduction

Mobile agents offer a new paradigm for trading on the internet. They are software applications that can work autonomously and have the capability to move from one device or platform to another to execute and perform tasks on different hosts for their owners in a wired and/or wireless network. A mobile agent provides a number of features such as flexibility (i.e. regardless of time and place), rapid and instant access (since the device is with the person and it is nearly always in connected mode of operation), and efficiency (i.e. speed). All such features apply to the variety of e-marketplace models (Yang, 2005). These devices, on the one hand, consume bandwidth only when they move but continue to execute after the move, even if they lose network connectivity with their principals: servers or communicating partners (Mahmoud and Yu, 2006). Using a mobile agent can reduce the network usage, reduce the dependency on network availability and avoid network latency (Alfalayleh and Brankovic, 2004). The device does not have to be connected all the time, thus saving on expensive network uptime and online mode of operations. On the other hand, it saves on processing time and usage of system resources. In addition, it can also operate as a group of sub-mobile agents in parallel processing mode over distributed sites to achieve greater efficiency (Panayionou *et al.*, 1999). Hence, a mobile agent-based e-marketplace can be regarded as a reliable and secure business model and platform for buyers and sellers to perform transactions of goods and services via the extended distributed channels over the internet or mobile network. Therefore, with further research and development of e-commerce models, they will enhance current models by using mobile agents, as pointed by Rodrigo and Stanski (2000).

1.1 Security issues, background and related works

Security at the application level covers various aspects, including authentication, authorization, message integrity, confidentiality, and overall management defense of the systems and its operation (Kannammal and Iyengar, 2007). For e-business applications, audit ability, non-repudiation and certification are added requirements. In addition, the use of mobile agents in e-commerce applications would impose other security threats such as masquerading, eavesdropping, malicious interception, spoofing, uncontrolled cloning, double spending, fraud and audit trail modifications by hackers and unethical system administrators. Thus, security mechanisms will have to be embedded to ensure that mobile agents will not sacrifice security requirements. Hence, mobile agents raise a number of security issues. Generally, it concerns two different types of security problems: one is the protection of the platform or host that runs the mobile agent against attacks, which can harm or use its resources without permission; the other is the essential protection necessary to guard mobile agents against the malicious hosts that might alter information it carries when it visits the hosts in its itineraries. Currently, there are several proposed methods such as sandboxing, authentication, authorization, proof-carrying code and payment check. But these methods are not enough to make the e-marketplace fully secured and trustworthy (Patel, 2010).

Mobile agent technology has its own limitations. The mobility characteristic of a mobile agent permits all the required operations to be performed locally in the e-marketplace without maintaining reliable connections with remote hosts and without bandwidth engagement. Generally, a mobile agent uses two mobile models: namely, single-hop and multi-hop models as shown in Figure 1.

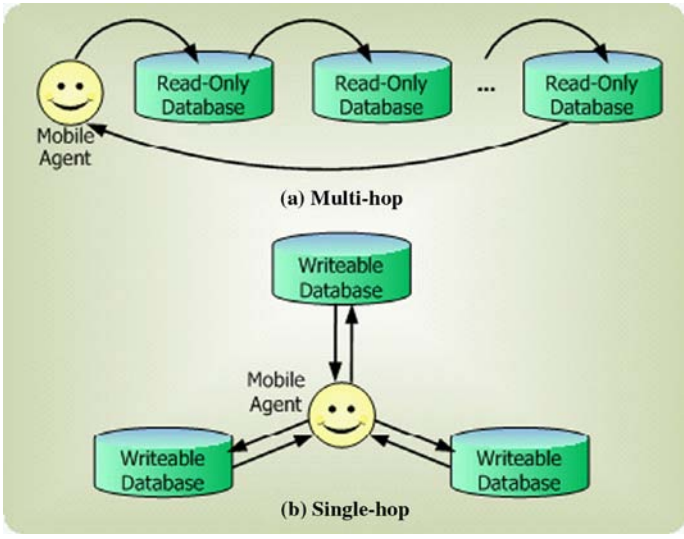


Figure 1.
Mobile agent in (a) multi-hop and (b) single-hop travelling scenario

In the e-marketplace environment, multi-hop is more efficient and useful than the single-hop. For instance, a mobile agent can search for a product in the lowest price range through the multi-hop to retrieve the message from each node. Nevertheless, the major weakness for this model is that every visited node may steal the sensitive information (such as e-cash and credit card number) or change the visited results to make a bias ranking of the prices. Therefore, in the mobile agent-based e-marketplace environment, strong mobility causes high security risks and threats, while weak mobility causes low security risks and threats. Figure 2 shows the relationship of security and mobility and indicates the position of a multi- and single-hop.

In order to reduce the security risks of a mobile agent, Zhang and Lin (2005) proposed a mechanism for e-commerce, which uses both single- and multi-hop to mitigate against any threatened risks. They use the multi-hop to collect the messages during the first phase of its operation. In this phase, the mobile agent takes the requirements and travels from node to node within the e-marketplace network infrastructure to collect the related information. The database on each node is a read-only database to protect itself against security threats and misuse. The mobile agent does not take any sensitive information with it when it performs information collection. After the mobile agent has visited all the nodes, it returns to its owner to report the results. During the second phase, the owner dispatches the mobile agent again to take the sensitive information to perform the e-business task, this time, by using the single-hop to reduce the security risks. Figure 1 shows how the multi- and single-hop travelling scenario operate in this case. Thus, the system security is improved by using this dual mechanism.

Mobile agents tend to raise a number of other security issues such as the protection of the e-marketplace network infrastructure and the platform/host that runs the mobile agent against malicious attacks, which can harm or use its resources without permission. Besides, the issue of protection of mobile agents and their supporting systems against malicious attacks from a variety of intervening sources, some attackers

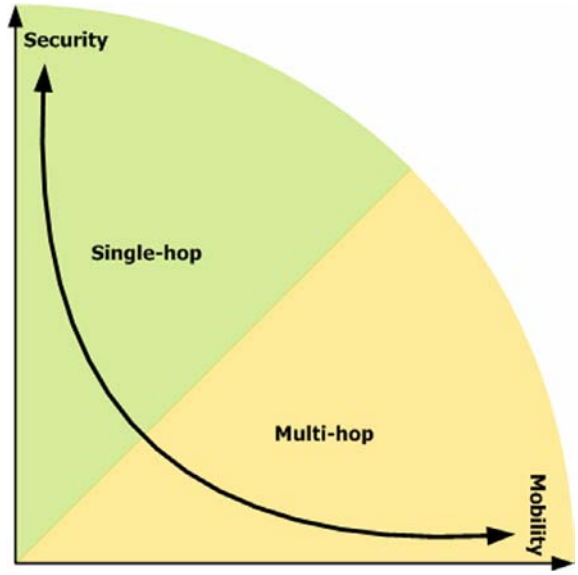


Figure 2.
Security – mobility with single- and multi-hop

might alter the information the agents carry and process it without their knowledge when they visit the hosts in their transactions' itineraries or schedules.

Safety measures, particularly, the security of mobile agents while recognized as the most important set of functions together with the techniques and protocols in the e-marketplace applications is still in its infancy and a major research topic. There are different security approaches for mobile agents that have been proposed to protect the platform, host, agents and route. For instance, the security protocols such as secure socket layer (SSL) (Freier *et al.*, 1996) and secure electronic transaction (SET) (Drew, 1999) of internet payment in a mobile computing environment are used for confidentiality and integrity to secure the communication between agents on different hosts. The SSL channel may not provide security for a mobile agent since a mobile agent may move to an insecure platform or host to communicate with other agents. Besides, it is dangerous for a mobile agent to exchange sensitive information without the use of cryptography techniques as the information can be stolen or corrupted by malicious attacking agents. On the other hand, SET offers better security than SSL since it uses public key infrastructure for privacy and X.509 digital certificates authenticates participants in the e-marketplace (Liu, 2003; Poggi *et al.*, 2003). More importantly, a buyer's sensitive information is not seen by the merchant, nor is it kept on the merchant's server to ensure buyer confidentiality, privacy and safety (Patel, 2010).

When mentioning security, audit and digital forensic techniques appear as major side issues (Antoniou *et al.*, 2008; Katos and Patel, 2008). These techniques are becoming very important because of the business trading auditing legal requirements and the rapid rise of cyber-crime activities on the internet. While digital forensics is mainly concerned with the admissibility of forensic evidence as set forth by the relevant court system, it is necessary to keep track of all pertinent transactions and information which can be used in an investigation for proofing of the criminal activities in the e-marketplace environment operated through the internet. It is very high on the

agenda of international law setting bodies and law enforcement agencies fighting against cybercrime (Cerezo *et al.*, 2007).

In the e-marketplace environment, real time digital forensic frameworks include several components such as data collection, data analyzing, evidence gathering and reporting and the supporting digital forensic protocols to complement the gathering of information tracking of illegal activities through monitoring and traps/honeypots. We can use specialist static agent(s) and mobile agent(s) to carry out these operations at a cost of systems management overheads, but in return it offers improved surveillance and intelligence, self-adaptability, flexibility and fault-tolerance in a distributed networked e-marketplace environment. Using a mobile agent to represent the digital forensic activities, it can automatically collect the network data from multiple distributed heterogeneous systems, ranging from event logging and packet capture. It can efficiently reduce the data storage requirements at a single monolithic system site whilst having the ability to execute the digital forensic program to process the data for subsequent transfer to a central site in a summary reporting form, thus reducing the bandwidth and communications overhead significantly (Patel, 2005).

The performance analysis is the most important issue for mobile agent-based e-marketplace to investigate its feasibility and justify its implementation. The goal of this paper is to evaluate the framework system of secure and trustworthy mobile agent-based e-marketplace (STMAE) with the corresponding experiment results.

1.2 Framework of a STMAE

Although several research works (Song and Korba, 2003; Zhang and Lin, 2005; Zhao *et al.*, 2007) have been dedicated to the design of agent-based marketplaces, the lack of standards for an agent-based e-marketplace framework reflects that there are still many issues that need to be resolved before a standard could be defined for such a framework (Jailani *et al.*, 2008; Patel, 2010). This also includes the associated protocols for mobile agents to perform e-marketplace activities, keep tracking illegal activities and record the digital evidence for the e-marketplace to make a STMAE environment. Therefore, to clearly investigate requirements and evaluation criteria for our research, we have reviewed the literature including infrastructure services, e-payment systems and safety measures of the mobile agent-based e-marketplace which included the security, privacy, safety and trust issues, the audit and digital forensic services (Patel *et al.*, 2010). Based on the investigated requirements and evaluation criteria, we proposed a framework of a STMAE with digital forensics (Qi and Patel, 2009). Figure 3 shows the framework of a STMAE.

1.3 System architecture

The STMAE system architecture and its components are designed according to the STMAE conceptual framework (Patel *et al.*, 2011). It is implemented as an object model following the object oriented development and programming paradigm consisting of the use cases, the use cases specifications, the development environment and tools. We applied the mobile agent technology with safety measure services in the client/server model for the system design and implementation. The implemented system is based on IBM Aglet (IBM, 2004), using Java programming language. Employing the Eclipse (Eclipse.Org, 2008) programming software enables the system to be integrated with other cross-middleware systems implemented on the same platform.

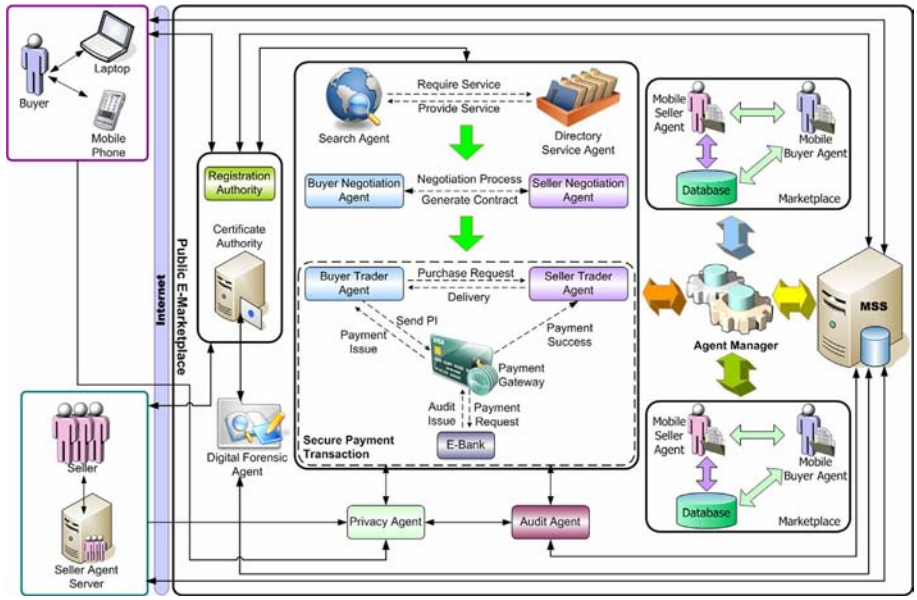


Figure 3.
Conceptual framework of a STMAE

The "Aglet: Tahiti Server" as shown in Figure 4 is presented as a mobile agent server which is used to identify components that run as applications, and what these system components are generically classified and called as a "marketplace" in the conceptual framework of STMAE (Figure 3).

The Aglets created for this marketplace program represent a "marketplace service server" as a server set. An Aglet called the agent manager (agent) is created on every Tahiti that represents a marketplace. Each marketplace loads a list of items per every

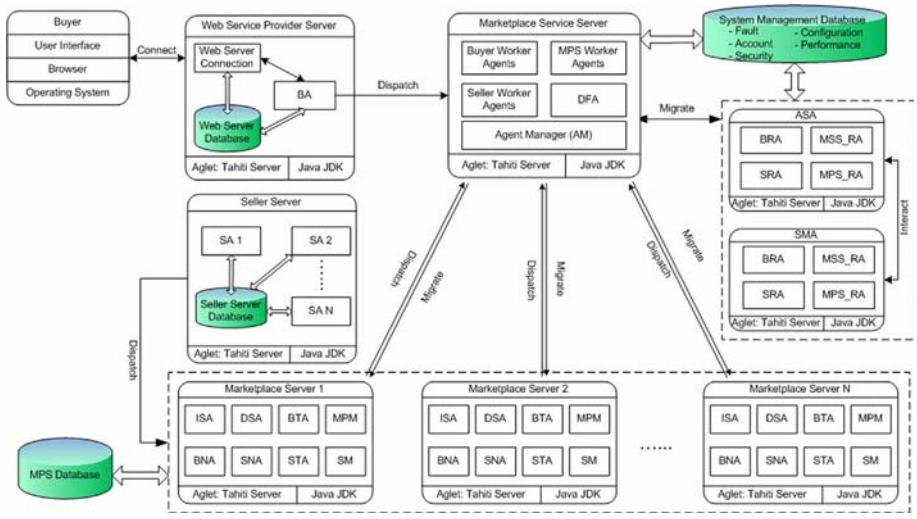


Figure 4.
System architecture of a STMAE

sub-component of that marketplace from a metadata file. Then the aglet created on the Tahiti server is the composite (marketplace service server (MSS)) consisting of all the sub-components representing that marketplace. The aglet agent launches a simple GUI which contains a search box and a list of buyer and seller sites through a set of IP addresses to perform e-business transactions. Thus, Tahiti is an application program that runs as an agent server. Aglets are created and controlled through a stand-alone Java server called Tahiti. Tahiti servers can create, send, receive and destroy Aglets. Figure 4 shows the system architecture of a STMAE mapping of components in the conceptual framework for system implementation.

According to Figure 4, the system architecture consists of at least four Aglet Tahiti Servers known as the MSS, the authority server (AS), the marketplace server (MPS) and the web service provider server (WSPS). MSS is the main service server in the public e-marketplace. The AS includes two different authority servers namely agent service authority (ASA) and security management authority (SMA). More than one MPS can exist in the public e-marketplace. WSPS provides web server service for a buyer to enter into the e-marketplace. The details of each Aglet Tahiti Server with its system components and responsibilities are described as follows.

Marketplace service server. The MSS is the main service server in the public e-marketplace that is used to handle the conversation between the buyer and the seller. It is responsible for maintaining the information for a set of marketplaces. To start up this server, it should first register at the AS to get the authorization for service providing. Thus, it is used to provide services for mobile agents from both the buyer and seller sides to perform the tasks such as searching for the information of the specified products and providing the offers of the products. There are several mobile agents that run on this service server, such as the agent manager (AM), buyer worker agents, marketplace worker agents, seller worker agents and digital forensic agents. Buyer worker agents are the general representations of a set of mobile agents that work for a buyer, such as the buyer registration agent (BRA), information search agent (ISA), buyer negotiation agent (BNA), and buyer trader agent (BTA). Marketplace worker agents and seller worker agents represent the RA of the MPS (MPS_RA), the RA of the seller (SRA) and the directory service agent (DSA). For those agents that run on this server, they can be divided into stationary agents and mobile agents. An AM is responsible for managing the work flow of mobile agents in the public e-marketplace. Therefore, an AM can be viewed as the stationary agent, while other agents that run on this MSS are all recognized as truly mobile agents. These mobile agents can migrate to different Aglet servers for task performing through the MSS. A system management database is used to record the successful transactions during the business process. It also records the log file of forensic investigations as an input into the evidential process. All sensitive log files are encrypted with a file hash and accessible in read-only mode.

Authority server. AS acts as a trusted third party (TTP) in the public e-marketplace. It runs two different authority servers, namely ASA and SMA. ASA acts as the registration authority which is responsible for the registration of all the participants in the public e-marketplace. All the participants should initially register at this AS when they want to participate into the public e-marketplace. The RAs (BRA, SRA, MPS_RA and MSS_RA) are presented here for registration. ASA will perform the verification of the particular certificate through the SMA. SMA acts as the certificate authority which is responsible for managing and generating certificates for all participants in the

system such as BA, SA, MSS and MPS. In addition, it is responsible for recording the evidence from preliminary investigations and auditing processes on participants in the e-marketplace according to any legitimate reported request for investigation of specific events. An authentication database which contains security attributes (such as certificate, etc.) is used for authentication and verification. This database also records the successful and unsuccessful authentication and verification of mobile agents.

Marketplace server. MPS consists of a set of marketplaces and seller agents that run simultaneously on the MPS. There can be more than one MPS in the public e-marketplace. Each of the marketplaces has the right to accept the registration and maintain a directory of seller agents. It also can authenticate incoming foreign mobile agents. This server is an execution environment for the incoming mobile agents such as ISA, BNA, BTA, DSA, SNA and STA. It can check that the visiting mobile agents are legal entities or illegal entities, therefore, to protect the sellers' information. In the MPS, marketplace manager (MPM) and security manager (SM) are the two stationary agents, while all the other agents are recognized as mobile agents. Several major components are involved in the marketplace as follows:

- *SM*. It is responsible for managing the security of participants in the public e-marketplace and their supporting mobile agents. It authenticates the credentials of incoming autonomous legitimate mobile agents as part of an overall e-marketplace and monitor the communication between the buyer agents and/or seller agents, and broadcast the sellers' certificates and security related information to other relevant servers (e.g. a registration process from the MPS to MSS).
- *MPM*. It is used to manage the marketplace for accepting the registration of a seller agent and cancellation of a trading transaction. It is also responsible for managing and maintaining the directory of seller agents in the DS.
- *DS*. It is responsible for the DSAs running on the server. The DSA maintains the seller's and product's information, which was recorded in the MPS database. DSA will periodically send the updated information to MPM for modifying the product's catalogue of the seller agents that are maintained in the DS.

Seller server. It is responsible for a number of seller agents running on the server. The seller agents that run on this server are responsible for dispatching worker agents to communicate with incoming buyer agents, provide the requested product information, monitor the execution of buyer agents and protect the local resources of the seller agents. At the same time, register the seller agents to the MPM and through it registers to the MSS when the seller is set up. It applies the certificate of the seller agent from SMA and sends it to the MPM. A seller server database is used to record all the information of sellers.

Web service provider server. A buyer should connect to the WSPS through the network connection. The buyer should register at this WSPS for the web service. The WSPS allows the buyer to create the mobile agent on behalf of the user at WSPS for the task performing in the mobile agent-based e-marketplace. The master buyer agent has the ability to create slave mobile agents at WSPS and distribute tasks to these slave mobile agents. This master buyer agent is the stationary agent running at this server. All the buyer worker agents are dispatched from this server.

The rest of the paper is organized as follows. Section 2 presents the importance of evaluation for mobile agents and mobile agent-based systems, and the experimental

settings for the system. Section 3 presents the evaluation of the system including performance, security, scalability, reliability, portability and modularity. Section 4 gives the discussion of the solution. Section 5 concludes the paper by discussing and highlighting the further research work.

2. Background research on evaluation

Evaluation of mobile agent-based systems is an important issue in today's world for the feasibility of software applications. As the businesses grow, the mobile agent-based systems should not only support their functions but also need to improve to support more users, process more data or both. Thus, it is important to maintain the performance of the systems (Smith and Williams, 2002). Therefore, performance and scalability can be considered as the most important issues in the system evaluation. However, the evaluation of the system is a difficult job. Currently, there are three methods for evaluation of performance namely analytical, real-system measurement and discrete-event simulation methods (Tursem *et al.*, 2003). Discrete-event simulation is a flexible, reproducible and an inexpensive method for performance analysis, especially for a mobile agent paradigm.

Mobile agents offer a new paradigm for trading on the internet since it offers advantages such as to reduce network usage, load sharing and delegation of a time consuming task, and so on. By proving such advantages, Spyrou *et al.* (1999) implemented their experiment in both fixed 10 Mb/s and wireless 9,600 b/s environment to compare the response times of query requests between a mobile agent and a client/server in both environments. The query requests include the first query and subsequent query. From their experimental results, the first query takes more time than the subsequent query; however, the mobile agents with interoperability always take less time than a client/server, especially in the wireless environment. Whereas, Braun (2003) developed a mathematical model for network traffic and response time of client/server based approaches and mobile agents for a general application scenario. The performance evaluation of network load between a mobile agent and a client/server is based on their mathematical model. Their evaluation confirms that the mobile agents produce fewer network loads than the traditional client/server techniques.

A mobile agent-based e-marketplace can be regarded as a good business model and platform for buyers and sellers to perform transactions of goods and services via the extended distributed channels over the internet or mobile network. Various factors can greatly affect the performance of mobile agent-based systems such as the agent implementation technique, execution environment, network load, agent system design principles, programming language specific properties and many other variables (Tursem *et al.*, 2003). For the agent implementation platforms, Fortino *et al.* (2008) identified the different issues for comparing the mobile agent frameworks that included three major aspects: comparison of the terminology and concepts, the main features of each agent framework, and the agent programming models. Another research work performed by Jha (2002) included the qualitative and quantitative comparison across three Java-based mobile agent frameworks namely Voyager, Aglets and Concordia, for e-commerce applications. According to the comparisons, agent services available in Aglets for the developer are: persistence, security, communication messaging, collaboration and web enabled agents. This platform is widely used as a test-bed for implementing agent-based systems. We apply the IBM Aglets as our mobile agent

platform for the system implementation. For the experimental settings, Kannammal and Iyengar (2007) developed a performance model in order to tune the parameters of the execution environment to meet the desired level of system performance by using two Pentium IV systems, each with 128 MB RAM with Windows XP as the operating system, and with both systems connected through 100 Mbps ethernet LAN.

In our system design, we have defined the development environment and tools for the system implementation and experiments. For the software, basically, the windows operating systems are suitable for the software development. We selected the Windows XP operating system for the development environment. For the hardware, the developer can either choose laptops or workstations. Furthermore, the system should be able to connect to the internet for both implementation and testing. The system is implemented in Java-based technologies applicable to a mobile agent system: Aglets as the system platform and Java programming for web-based system to realize prototype e-marketplace system with XML and some security components. The work was to implement the system in such a manner, so that it is compatible with different server platforms. The Apache Tomcat is selected for displaying the system results. It is a non-commercial product that provides reliable and robust features to serve the purpose and it is a standard servlet engine implementation. Java JDK was selected as the platform because it is widely available on a number of servers.

In our system implementation, we implemented the business process of the system and partially implemented the e-secure payment system and digital forensics. Here, for the e-secure payment system, we describe the process of payment for our system to demonstrate how the payment would be implemented. The payment starts from the cardholder who is the buyer. The cardholder initially sends the purchase request to the seller. The seller then accepts the confirmation of the purchase. After that, the cardholder sends the payment information to the issuer through the payment gateway. The issuer is the bank where the cardholder opens his/her account. The issuer transfers payment through the payment gateway to the acquirer which is the bank with which the supplier opens his/her account. When the acquirer receives the correct payment from the issuer, it sends the payment confirmation to the payment gateway. The payment gateway then sends the payment confirmation to both the cardholder and the supplier. Hence, the payment process has been accomplished and the goods despatching and delivery starts. The cryptography technique used is embedded in the payment process to protect the confidentiality of the payment data, therefore, ensuring a secured payment process. For the digital forensic functions as stated above, we implemented a digital forensic agent currently working as a real time evidence gathering recorder in our system that can be enabled to capture all mobile agent transaction activities and the network packets pertaining to its data transmission activities, which are logged in an encrypted file with the application of digital signatures and time stamps to ensure non-tampering. This log file is available as input for analysis purposes during a formal investigation of evidence production and presentation. It can also be used by the system or network administrator to view this log file to see who has visited the server for a variety of other system management reasons. Due to the implementation of these functions, the functionality and system performance has been analysed and evaluated from experimental pilot operation runs, observations, data sets and outcomes against the evaluation criteria established during the initial requirements capture stage of the research work.

3. Evaluation

The requirements and evaluation criteria are used to formulate the proposed framework and system. They are also the foundation of our system design, implementation and evaluation. The functional requirements are defined and classified into three major areas:

(1) The requirements for infrastructure services, the requirements for safety measure services and the requirements for technical supports for both design and implementation.

(2) The requirements of infrastructure services indicate the basic services which are known as business processes of the framework and system.

(3) The requirements of safety measure services are security techniques to protect the e-marketplace which has the ability to provide a secure environment for the buyer and the seller to trade goods.

(4) The last requirement is the technical support for design and implementation which covers the software and hardware for the system implementation and the marketplace environment for users.

As we pointed out previously, that performance and scalability are the two most important issues for mobile agent-based systems with their feasibility. Thus, we now focus on the evaluation of the non-functional behavior of the framework and system which includes six evaluation criterions as shown in Table I. Each of the criterions will be evaluated in the following subsections.

3.1 Performance analysis of searching mechanism

The performance analysis of the searching mechanism meets the evaluation criteria C1 in Table I. The performance of the system is an important factor for the overall usability

No.	Criteria	Description
C1	Performance	The system should have good performance. Response times of the mobile agent-based e-marketplace should vary between reasonable upper and load bound threshold times for each of the services provided by the system. Performance tests should be carried out fine tune the system
C2	Scalability	The system should be scalable. The scalability tests should be carried out which include the response times and throughputs of the system as the number of agent clients and servers are increased
C3	Security	The system should provide the security model for the system to make a secure environment for participating in an e-marketplace trading environment
C4	Reliability	The system should operate 7×24 hours every week of the year to provide available services for the users on an optimum basis
C5	Portability	The system should provide the portability of the agent clients and servers under other operating system platform for various devices
C6	Modularity	The system should be implemented using as many standards as possible, to provide a flexible and easy way for further development on interfaces and mechanisms in order to operate in an autonomous cooperative federated e-marketplace trading environment The system should be built in a flexible and extensible manner to facilitate an e-marketplace system's business logic

Table I.
Evaluation criteria of STMAE

of the system. The size of the result data is normally small. Therefore, it may take a longer time for the mobile agent to complete its tasks when a large number of data are transferred back in different time periods. However, it may easily cause a bottleneck from the client side for a number of mobile agents to be dispatched for performing a specific task, such as searching for a specific product. In this matter, the serial dispatch takes a longer time to retrieve the results. With the theoretical analysis, the dispatch time for the searching mechanism can be obtained. Suppose t is the time for dispatching a mobile agent. If a number of mobile agents n ($n \geq 2$) should be dispatched, b ($b \geq 2$) is the branch number of the search tree, and $h = \log_b n$ ($h > 1$) is the height of the search tree, then, the total dispatch time is $T_S = h*b*t$. It can be proven by the deduction of h, which is the height of the search tree. For $h = 1$, the master agent (MA) should use time b*t to dispatch "b" branch of slave agent's (A). Let the search tree be a "b" branch dispatch tree of height "h" ($h > 1$). The "MA" becomes the initial layer (L_0), then, $L_{h-1,n}$ is the last agent to be dispatched at the same layer of the search tree where the height is $h - 1$. Thus, the total time from the beginning to the time when the $L_{h-1,n}$ is dispatched is $T_S = (h - 1)*b*t$. If $L_{h-1,n}$ is a search agent (SA), then it should dispatch "b" branch of "A"s. The time taken for it is "b*t". Since there is a virtual dispatch in the search tree, therefore, it can save $(h - 1)*t$ dispatch time. Finally, the total time for dispatching all numbers of "A"s should be:

$$\begin{aligned} T_S &= (h-1)*b*t + [b*t - (h-1)*t] \\ &= (h-1)*b*t + [(b-h+1)*t] \\ &= [(h-1)*b + (b-h+1)]*t \\ &= (h*b - h + 1)*t \end{aligned} \tag{1}$$

Here, we propose three test models to evaluate the performance analysis. These test models are described as follows:

- *Model 1*. Searching for a specific product from one marketplace.
- *Model 2*. Searching for a specific product from two distributed marketplaces.
- *Model 3*. Searching for a specific product from four distributed marketplaces.

These three models can dispatch the ISA to the e-marketplace for a specific product search from every marketplace based on the proposed searching mechanism which integrated both single- and multi-hop (Qi, 2011). The length of the XML file is set to 1,000 records, 10,000 records, 100,000 records and 1,000,000 records. Table II shows the execution time of searching for a specific product in different situations with different models from the marketplaces, where Table III shows the average time taken of each model. According to the results listed in these two tables, Figure 5 shows the performance test results of the three test models for searching a specific product from

Table II.
Execution time of searching a specific product in three different models

Test	No. of records	Model 1 time taken (ms)	Model 2 time taken (ms)	Model 3 time taken (ms)
1	1,000	16	31	47
2	10,000	125	147	171
3	100,000	656	748	850
4	1,000,000	6,813	7,438	8,072

XML file, Figures 6 and 7 show the line chart of the time taken for the three different test models and the average time taken of each test model, respectively.

By comparing these three models in reading the same data, the time taken is increasing marginally (slightly) in each test with the same records. The average time taken is also increasing slightly from Models 1 to 3. Therefore, the time growth rate of M_A and M_B is calculated based on the experimental results as shown in Tables II and III. The growth rate of M_A means the time growth rate from Models 1 to 2, whereas the growth of M_B means the time growth rate from Models 2 to 3. The formula of the growth rate calculation is:

$$R_G = (T_{M(n+1)} - T_{M(n)})/T_{M(n)} * P_H \quad (2)$$

where R_G stands for a time growth rate, T_M stands for time taken of the tested model, P_H is noted as a static value – 100 percent and n ($n > 1$) is the number of the tested model. Table IV shows the results of the time growth rate calculation and it is graphically shown in Figure 8.

From Table IV and Figure 8, to search for a specific product from 1,000 records of each marketplace, the growth rate of M_A was 93.8 percent. The growth rate of MA sharply decreased to 17.6 percent when searching for a product from 10,000 records of each marketplace. It then gradually decreased to 9.9 percent when the length of data sets became 1,000,000 records. M_B had the same situation as M_A. The growth rate was initially at 51.6 percent but sharply decreased to 16.3 percent when searching from 10,000 records, then continued to decline until 7.9 percent when searching for the product from 1,000,000 records of every marketplace. As described, the growth rate gradually declined from 1,000 to 1,000,000 records indicated that the growth rate decreased with

Models	Marketplaces	Average time taken (ms)
Model 1	1	1,902.5
Model 2	2	2,091
Model 3	4	2,273.75

Table III.
Average times taken of three models for searching a specific product

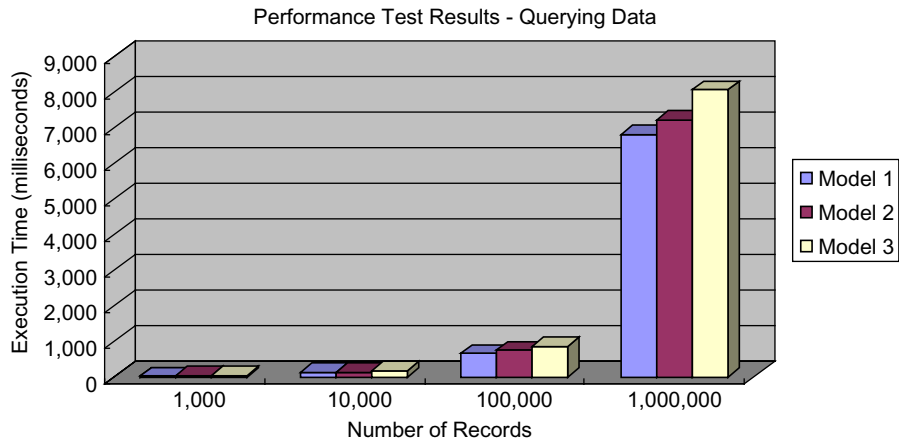


Figure 5.
Performance test results for searching a specific product

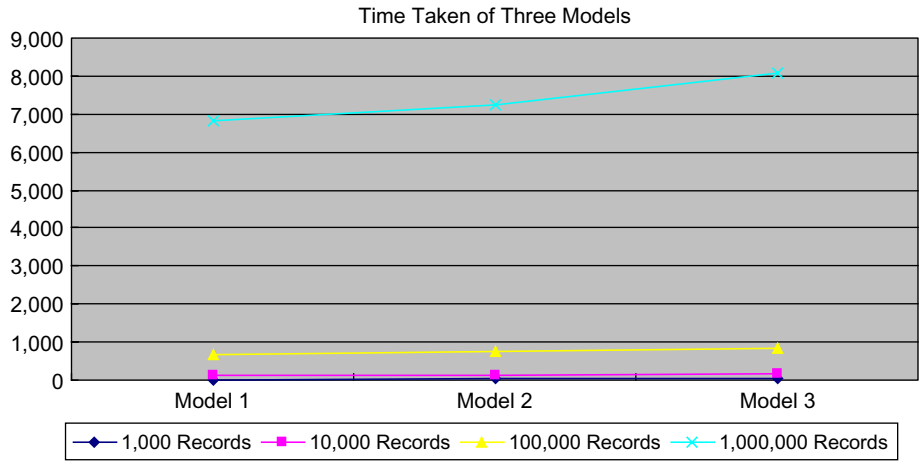


Figure 6.
Times taken for searching a specific product by the three different models

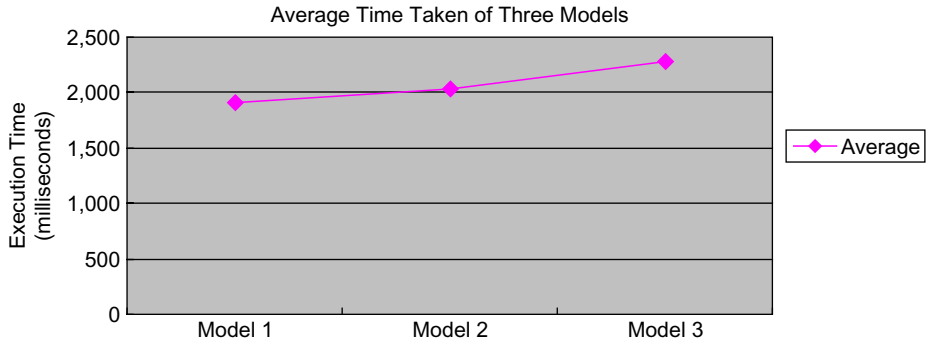


Figure 7.
Average times taken for searching a specific product by the three different models

Test	No. of records	Growth rate of M_A (%)	Growth rate of M_B (%)
1	1,000	93.8	51.6
2	10,000	17.6	16.3
3	100,000	14.0	13.6
4	1,000,000	9.2	7.9
5	Average R_G	9.9	8.7

Table IV.
Results of time growth rate calculation

increasing length of records. Especially, when the search agent searched for a product from 1,000,000 records, the growth rate of both models was less than the average growth rate. These numerical results demonstrate that the larger the records, lesser was the time growth rate taken for product searching. On the other hand, the growth rates from M_A to M_B also declined with the increasing number of distributed e-marketplaces. From the analysis above, it showed our searching mechanism was efficient when searching for a large number of data sets from distributed e-marketplaces in the network environment. It was a fundamental feature of searching for a specific product from a huge number of e-marketplaces on the internet.

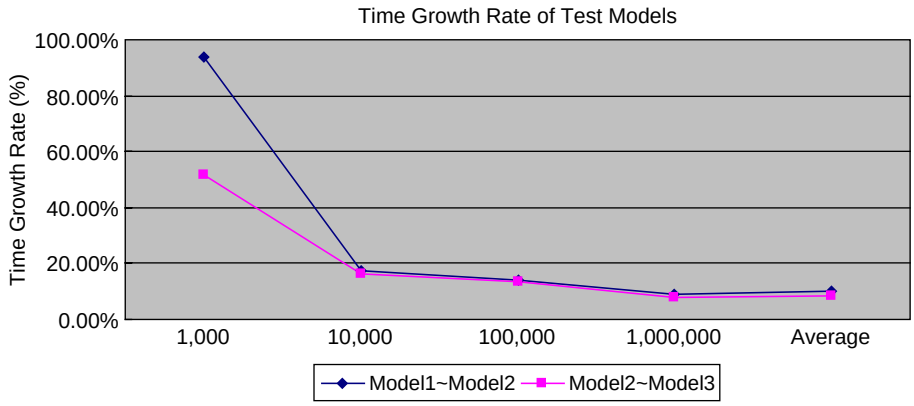


Figure 8.
Time growth rate of the tested three models

3.1.1 Results analysis of negotiation process. In the automatic negotiation process, the BNA obtains the results from the seller agents and outputs the offer price. In a certain negotiation round, if the new offer proposed by the buyer is rejected by most of the sellers for the first time, it means the proposed offer is too aggressive for the sellers. Likewise, if the offer proposed by the buyer is accepted by all the sellers, this means the offer is less aggressive. Thus, an adjustment factor is used during the automatic negotiation process. An experimental test was carried out to fit in the second case where the offer proposed by the buyer is accepted by all the sellers in the first round. The buyer decreases the price until it meets the desired price or the round limit is reached. Suppose that the lowest offer price is $2,038 in the offers list after the offer evaluation. The round limit of the negotiation is defined as five rounds. The priority of this negotiation sample is based on the seller's security level and trading reputation (determined during initial registration with the e-marketplace and subsequently adjusted during the seller's life time against pre-established security and business rating criteria), and the price of the product that the seller agent offers. The buyer proposes a price that is decreased by 2 percent, which is $1,975. If one of the seller agents rejects this offer, the negotiation agent will then negotiate with other seller agents. The price is then further decreased by the negotiation agent by another 2 percent based on the current price which is $1,935. If all the sellers reject this price, the negotiation process will stop at this round. Otherwise, the price can be continuously decreased in each round until it reached the round limits. The negotiation process as a series of steps is illustrated in Table V.

Round no.	Price in $ (decreased 2%)	Seller agent 9 ($2,038)	Seller agent 6 ($2,143)	Seller agent 8 ($2,891)
1	1,975	Accept	Accept	Accept
2	1,935	Accept	Accept	Reject
3	1,896	Accept	Accept	–
4	1,858	Accept	Accept	–
5	1,820	Reject	Accept	–

Table V.
Results of the negotiation process

3.2 Scalability

Scalability is one of the most important criteria in a system evaluation. It meets the evaluation criteria C2 as declared in Table I. However, the number of users and the frequency of the usage cannot be determined and is fixed beforehand. An e-marketplace application has two approaches to reach the goals. On the one hand, it is the software scalability and on the other hand it is the hardware scalability. The goal of these techniques is to extend the system resources, meaning the software components and hardware facilities to adapt to every possible number of users. The approach taken into evaluating the scalability of the implementation was to evaluate two factors. The first was the behavior of different response times of the system (the total response time, the average response time and the median time taken) as a function with a number of simultaneous clients. The second was the behavior of the throughput of the system as a function with a number of simultaneous clients. Table VI shows the results of response times and throughput of our implemented prototype. Based on these summarized results, Figure 9 shows the response times and load of the system and Figure 10 shows the throughput per second of the system.

Table VI.
Results of response times and throughput of STMAE

Test	No. of clients	No. of requests	Total response time (ms)	Average time taken (ms)	Median time taken (ms)	Throughput response/sec
1	1	10	104	41	49	19.20
2	5	50	197	125	130	23.27
3	10	100	368	276	279	30.40
4	20	200	662	453	457	36.54
5	30	300	957	677	688	36.82
6	40	400	1,315	985	1,020	37.46
7	50	500	1,748	1,307	1,392	38.71
8	70	700	2,711	1,857	1,889	36.10
9	90	900	3,772	2,135	2,258	34.31
10	100	1,000	4,741	2,553	2,575	33.81

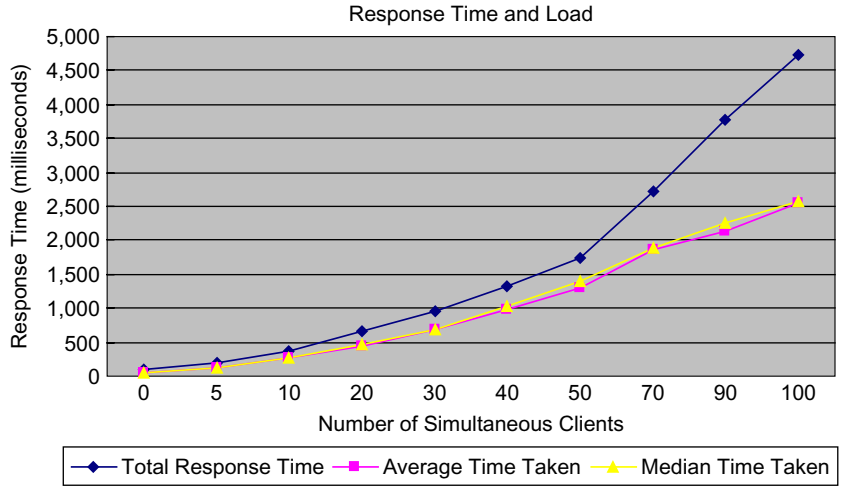


Figure 9.
Response times and load of system

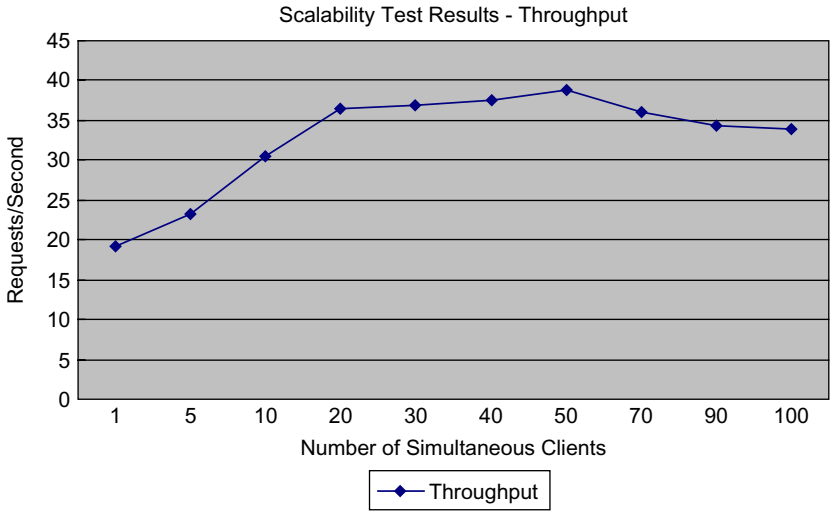


Figure 10.
Scalability test results – throughput of STMAE

As shown in Table VI, we performed ten experiments in this scalability testing. The number of simultaneous user clients was tested from 1 to 100 and the number of the requests was tested from 10 to 1,000. On the relevant line chart shown in Figure 9, the average response time seems to be increasing linearly from 1 to 50 simultaneous active clients; the slope of the curve is starting to steepen between 50 and 70 clients and the average response time gradually increasing from 70 to 100 clients. It is quite natural that every system reaches the level of load where the performance observed by a single user starts to degrade dramatically. For our system, the point of the simultaneous user clients seems to be somewhere above 65 but nearer 70.

As shown in Figure 10, the throughput increases rapidly up to five simultaneous user clients and continues increasing up to 50 users. The system has already reached the saturation point at 70 simultaneous users, and the throughput is now on the decline at this point. The throughput decreases from 70 to 100 users. Obviously, as shown in the figure, the system starts to become congested somewhat above 70 simultaneous users. When that point is reached, the increase in response time experienced by a single user is reasonable and linear as a function of load. Generally, the system fulfils the criteria set for reasonable scalability. In the experiments testing the spectrum of load, the system scales well. Based on the throughput of the system, we can calculate the daily transactions of the system. The formula of a transaction calculation is:

$$T_D = T_S * H_D * M_H * S_M \tag{3}$$

where T_D stands for transactions per day, T_S stands for the transactions per second, H_D is noted as hours of a day, M_H is noted as the minutes per hour, and S_M stands for seconds per minute. The system will flexibly support up to 3,283,200 transactions per day on the hardware that was used for the tests. Thus, the system throughput exhibits a very scalable behavior.

3.3 Security

Security meets the evaluation criteria C3 as declared in Table I. According to the security requirements that are described in the requirements' paper for the system implementation, several security mechanisms are provided by the system. Verification of the absolute fulfilment of all the security requirements is difficult, if not impossible. Meeting the criteria is analysed here by discussing the decisions taken and the ways in which the security threats were mitigated. Below, this discussion is presented for each of the core security requirements. For data transmission security, meaning securing the mobile agents communication between clients and the system, the e-marketplace has to offer an encryption mechanism such as cryptography for secure data transmission. Both the architectural design and the system implementation fulfill the requirement of preventing unauthorized entities accessing the mobile agent-based e-marketplace. In the architectural design, it is possible to isolate the system effectively from the network by protecting it with the safety measure protocols and to address filtering configurations. The safety measure protocols are defined and described in detail for the architectural design (Qi and Patel, 2009). In the mobile agent system implementation, the safety measure protocols (secure migration protocol) and authentication mechanisms have been implemented to meet the security requirements. However, the encryption and authentication mechanism for data transmission in the secure and trustworthy e-marketplace infrastructure is based on the cryptography technique which is independent and transparent to the application. Users must go through the TTP to verify their identity before they are authorized to use the system. Other safety measure protocols such as the secure payment protocol is partially implemented as a simulated prototype function since we did not have access to any e-payment fund transfer facility. The digital forensic evidence gathering recorder function that we have currently implemented in the system can capture all mobile agent transaction and network transfer activities and save them in a log file as a journal recording for subsequent investigation analysis and evidence presentation of system events.

3.4 Reliability

Reliability meets the evaluation criteria C4 as declared in Table I. For the prototype, we assumed a highly fault-tolerant distributed hardware environment treated as a single mobile agent platform. So, when the system runs on a single mobile agent platform within this environment, automatically, replication takes place. The required demand for reliability is fulfilled since every component of the system exists at least twice, and a failed component is taken over by its mirrored component and executes over multiple access points to the marketplace. By using this, first, the single points of failures are prevented, because if one server or component fails another can take over. Second, it opens the ability for maintaining parts of the system, while other parts are active. Therefore, when the components need to be maintained, it can be switched off and substituted with other active parts. The new server recovers the session data from the database, because it executes business logic for generating the response. Finally, the reliability of the database can be increased by using hardware clustering.

3.5 Portability

Portability meets the evaluation criteria C5 as declared in Table I. For the efficiency and portability of the STMAE system, it integrates the search mechanism by integrating

both single- and multi-hop. By using this dual mechanism, it not only improves the efficiency of the system but also reduces the security threats when it performs tasks in mobile clients and servers in an e-marketplace environment. The mobile devices (laptops, PDAs or mobile phones) capable of hosting STMAE components as mobile agents running complex applications under their respective operating systems were a prerequisite requirement. Today, perhaps the most exciting and rapidly changing area of systems and application software development is in the smart phone or multi-mobile domain. The latest techniques and simple design methods to develop and build applications to run on native iOS, Android 2.2 OS, BlackBerry OS, and Windows Mobile 6.1 platforms can be easily achieved by running JavaScripts and Java object code programs. It is possible to establish on these operating systems a Java virtual machine. Since the STMAE is implemented by using generic mobile agent software technology, Java programming language and Java-based platforms (IBM Aglet), this means the system is 100 percent Java based. It employs the mobile agent, and offers the mobility that can be executed on a standard Java platform, thereby achieving the portability. As an experiment, we ported it to the Android 2.2 OS and Windows Mobile 6.1 platforms with relative ease. Thus, we believe that it can be easily ported and run on any of the abovementioned operating systems.

3.6 Modularity

Modularity meets the evaluation criteria C6 as declared in Table I. The implemented system is structured in a modular way according to our current understanding. However, it is impossible to prove this conclusively. Evidence for this can be produced. Here, we discussed it from different aspects. In our STMAE system, the system architecture is based on a layered design paradigm to assist modularity. The business logic is separated from the user interface logic which is further separated from the visual layouts and data presentations. The business logic of the system is implemented using a modular-based approach of separation of concerns for given sets of functions. Moreover, the object classes such as the authorization class, seller class, or the server class that may be exchanged are not bound in programs by other third party software. The protocols for safety measures are implemented as services to STMAE components. The STMAE is designed very flexibly to allow various experiments, tests and evaluations of different mechanisms and protocols to take place from different viewpoints. Thus, modularity with clearly defined interfaces to achieve a very high level of modularity criteria to permit plug and play of different transactions, protocols and activity classes during the development stages are very important before the final system is put into operation. It also allows for growth and maintenance of the system in a more natural way.

4. Discussion

In this paper, we have discussed the evaluation of the non-functional behavior of an implemented system from the above sections. Some strengths and limitations of the solution are discussed here. Our system is implemented by using Java programming and an Aglet platform. The Java-based mobile agent technology enables the features of light-weighted, portable and platform independent clients. The architecture is suitable for both mobile and stationary users. In our system, a buyer only needs to input information for the product and fills up the preferences through a web browser.

Then, the buyer's requests will be distributed to the master mobile agent at a local place and this master mobile agent in-turn distributes the tasks to the slave agents who are dispatched for subsequent processes such as product searching, offer comparison, and negotiating. These processes are totally transparent to the end-users. The benefit on the performance of our solution is that the proposed architecture can easily support both single- and multi-hop processing by using mobile agents, as with our searching mechanism. The master mobile agent can send a number of slave mobile agents for searching and negotiating models to fulfill request tasks from buyers in parallel. Sometimes, the serial migration of a mobile agent offers the flexibility for the buyer in some specific request. However, our system provides some of the features by using the proposed searching mechanism (Qi, 2011) that offers a high efficiency on product searching and provides the support to help a buyer to find the best deal from a number of distributed mobile agent-based e-marketplaces. Our framework and system also provides cryptographic techniques and safety measure protocols for participants to trade goods in a secure and trusted environment. The service authority and security authority implemented in our system acted as trusted third parties that were used to identify and authenticate whether the buyers were legal or illegal entities. The reputation of sellers combined with their security level was implemented in the system. The reputation and security level of the suppliers can be upgraded through successful business transactions with the buyer in the mobile agent-based e-marketplace. However, we partially implemented the services of a secure payment and digital forensics. As we were unable to measure these two services, this could be viewed as the drawback of our solution, but we will continue to research this drawback.

5. Conclusion and future work

Despite the security weaknesses of mobile agent-based e-marketplaces' activities (Patel *et al.*, 2010), the future of mobile agent systems can be seen as promising (Zhao *et al.*, 2007) with proper embedding of security and other safety measures. However, more research is required to carve out a definitive framework and architecture that can be universally used to create and promote both *de facto* and international standards to achieve this as well as interoperability. Many security mechanisms have been proposed in order to improve mobile agent security. Although the complete answer to the security problems could not be found, mobile agents can be used in many tasks using today's technology to facilitate e-marketplace trading with some obvious risks and restrictions. The success of mobile agent technology in the future will depend on resolving the security and safety measure issues.

The result of our research work was that we designed and implemented a STMAE with safety measure services. Accordingly, the outcomes and achievements of the research have been described here and elsewhere. We reviewed the current literature with an up to date state-of-the-art mobile agent-based frameworks and systems, including the infrastructure services, e-payment systems, the current status of secure mobile agent-based e-marketplaces and safety measures of the mobile agent-based e-marketplaces which included the security, privacy, safety and trust issues, the audit and digital forensic services (Patel *et al.*, 2010). Based on the reviewed literature, we investigated the requirements and evaluation criteria that were used to formulate the secure and trustworthy mobile agent-based framework and safety measure protocols. The proposed framework with the safety measure protocols were designed and specified,

in which the safety measure protocols included secure migration protocol, secure payment protocol and digital forensic protocols as defined and specified (Qi, 2011). The system design for the system implementation was discussed from several aspects, such as UML and modeling tools, class diagram, use case diagram, use case specifications, and we also discussed the development environment and corresponding tools (e.g. Aglet, Eclipse, Apache Tomcat, XML and Jpcap Library). A prototype was implemented in order to analyze, evaluate, verify and validate the proposed framework. However, it would require a greater effort to implement a comprehensive prototype. The implemented prototype employs mobile agents including infrastructure services and business processes with partial safety measure services which offered several features such as easy to use for end-users, easily configurable, provides high efficiency for product searching, reduces risks of security threats, provides automatic negotiation process between buyers and suppliers, and records packets in a log file as evidence, in a secure trading environment. The secure payment process and digital forensic service were partially implemented in our prototype. We will concentrate on these two services in our future research. We have evaluated the solution in this paper against the requirements and evaluation criteria based on the analysed results from experimental runs, observations, data sets and outcomes. Every aspect of the research was addressed as comprehensively as possible. The prototype can be used as a basis in future implementation for a real-life system. From our research work, we conclude that the following items of research and development still need to be performed:

- Enhance the user interface with intelligent features that can be used in different mobile devices such as simple to smart PDA.
- Experiment with different cryptography algorithms for those aspects of the safety measures in order to measure the performance and benefits and to show that they fulfill the functionality of STMAE e-trading.
- Verify and validate the safety measure protocols to be correct and reliable.
- Implement the payment processes with the proposed secure payment protocols to make the payment scheme more transparent and secured under various cybercrime and hacking conditions.
- Define new safety measure protocols (e.g. e-secure payment and digital forensics, etc.) for e-market trading that can be universally used to create and promote both de facto and international standards to open up the whole e-economy.
- Make the implementation generic to allow for "plug and play" of new protocols and services to be tested and it can be automatically reusable for further research and system development.
- Change or upgrade the developing tools for system development to enhance the usability and reusability of the system if new techniques appear in future.

Mobile agents can be used in many tasks using today's technology to facilitate e-marketplace trading with some obvious risks and restrictions. The success of mobile agent technology in the future will depend on resolving the security and safety measure issues. The authors have partially solved the security problems. We hope

further research can shed more light and help improve the worthy of our proposed framework and system in real operational environments and encourage more research elsewhere.

References

Alfalayleh, M. and Brankovic, L. (2004), "An overview of security issues and techniques in mobile agents", *Proceedings of the Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security* (*CMS2004*), Vol. 175, pp. 59-78.

Antoniou, G., Leon, S., Stefanos, G. and Paramalli, U. (2008), "Privacy and forensics investigation process: the ERPINA protocol", *Journal of Computer Science and Interface*, Vol. 30 No. 4, pp. 229-36.

Braun, P. (2003), "The migration process of mobile agent: implementation, classification, and optimization", PhD disseration, Computer Science Department, Friedrich-Schiller-University Jena, Jena, available at: www.db-thueringen.de/servlets/DocumentServlet?id=1059 (accessed 2 May 2009).

Cerezo, A.I., Lopez, J. and Patel, A. (2007), "International cooperation to fight transnational cybercrime", *Second International Workshop on Digital Forensics and Incident Analysis* (*WDFIA 2007*), pp. 13-27.

Drew, G.N. (1999), *Using SET for Secure Electronic Commerce*, Prentice-Hall, Upper Saddle River, NJ, p. 265.

Eclipse (2010), Eclipse documentation – Current Release. Eclipse Galileo, available at: http://help.eclipse.org/galileo/index.jsp?topic=/org.eclipse.platform.doc.isv/guide/int_eclipse.htm (accessed 20 February 2012).

Fortino, G., Garro, A. and Russo, W. (2008), "Achieving mobile agent systems interoperability through software layering", *Information and Software Technology*, Vol. 50 No. 4, pp. 322-41.

Freier, A.O., Karlton, P. and Kocher, P.C. (1996), *The SSL Protocol Version 3.0*, available at: http://home.netscape.com/eng/ssl3/ssl-toc.html (accessed 27 August 2008).

IBM (2004), "Aglet workbench", available at: www.trl.ibm.co.jp/aglets/ (accessed 20 February 2012).

Jailani, N., Yatim, N.F.M., Yahya, Y., Patel, A. and Othman, M. (2008), "Secure and auditable agent-based e-marketplace framework for mobile users", *Journal of Computer Science & Interface*, Vol. 30 No. 4, pp. 237-52.

Jha, R. (2002), "Mobile agent for e-commerce", Master thesis, KR School of Information Technology, Indian Institute of Technology, Bombay, available at: www.it.iitb.ac.in/~sri/students/rahul-thesis.pdf (accessed 23 April 2009).

Kannammal, A. and Iyengar, N.Ch.S.N. (2007), "A model for mobile agent security in e-business applications", *International Journal of Business and Information*, Vol. 2 No. 2, pp. 185-98.

Katos, V. and Patel, A. (2008), "A partial equilibrium view on security and privacy", *Information Management & Computer Security*, Vol. 16 No. 1, pp. 74-83.

Liu, D.S. (2003), "Research of the two electronic commerce payment protocols: SSL and SET", *Security and Safety Magzine*, Vol. 4, pp. 61-3.

Mahmoud, Q.H. and Yu, L. (2006), "Making software agents user-friendly", *Computer*, Vol. 39 No. 7, pp. 94-6.

Panayionou, C., Samaras, G., Pitoura, E. and Evripidou, P. (1999), "Parallel computing using Java mobile agents", *Proceedings of 25th EUROMICRO Conference, Milan, Italy, 8-10 September*, Vol. 2, pp. 430-7.

Patel, A. (2005), "An automatic computing approach to developing secure, trusted and auditable services for e-business", *International Journal of Computer Systems Science & Engineering*, Vol. 20 No. 6, pp. 433-7.

Patel, A. (2010), "Concept of mobile agent-based electronic marketplace – safety measure", in Lee, I. (Ed.), *Encyclopedia of E-Business Development and Management in the Digital Economy*, IGI Global publications, Western Illinois University, Macomb, IL, pp. 252-64.

Patel, A., Qi, W. and Taghavi, M. (2011), "Design of secure and trustworthy mobile agent-based e-marketplace system", *Information Management & Computer Security*, Vol. 19 No. 5, pp. 333-52.

Patel, A., Qi, W. and Wills, C. (2010), "A review and future research directions of secure and trustworthy mobile agent-based e-marketplace systems", *Information Management & Computer Security*, Vol. 18 No. 3, pp. 144-61.

Poggi, A., Tomaiuolo, M. and Vitaglione, G. (2003), "Security and trust in agent-oriented middleware", in Meersman, R. and Tari, Z. (Eds), *OTM Workshops 2003*, Vol. 2889, Springer, New York, NY, pp. 989-1003.

Qi, W. (2011), "Design and implementation of a framework system architecture for secure and trustworthy mobile agent-based e-marketplace", Master thesis, Department of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi.

Qi, W. and Patel, A. (2009), "A secure and trustworthy framework for mobile agent-based e-marketplace with digital forensics and security protocols", *International Journal of Mobile Computing and Multimedia Communications* (*IJMCMC*), Vol. 1 No. 3, pp. 8-26.

Rodrigo, T.D. and Stanski, A. (2000), "The evolving future of agent-based electronic commerce", in Rahman, S.M. and Raisinghani, M.S. (Eds), *Electronic Commerce: Opportunity and Challenges*, Idea Group Publishing, Hershey, PA, pp. 337-51.

Smith, C.U. and Williams, L.G. (2002), *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*, Addison-Wesley, Boston, MA.

Song, R. and Korba, L. (2003), "Security communication architecture for mobile agents and e-commerce", *Proceedings of the 2003 International Work shop on Mobile Systems, E-Commerce and Agent Technology* (*MSEAT'2003*), *Miami, Florida, USA*.

Spyrou, C., Samaras, E. and Evripidou, P. (1999), "Software models for wireless and mobile computing: survey and case study", Technical Report TR-99-5, University of Cyprus, Nicosia.

Tursem, M.E., Guneş, M.H., Yildiz, M. and Kuru, S. (2003), "Performance analysis of mobile agents using simulation", *Proceedings of the Advanced Engineering Design Conference* (*AED2003*), *Prague, Czech Republic*.

Yang, X.F. (2005), "Mobile agent computing in electronic business: potentials, designs and challenges", PhD thesis, School of Information Technology, Griffith University, Gold Coast.

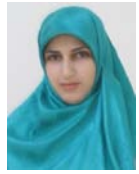
Zhang, D.L. and Lin, C. (2005), "Security model of mobile agent in e-commerce", *China Academic Journal Electronic Publishing House*, Vol. 25 No. 6, pp. 1271-3.

Zhao, S.H., Xin, F.Q. and Ma, J.Z. (2007), "Research on secure mobile agent-based electronic commerce", *Journal of Science and Technology Information*, Vol. 2, pp. 10-11.

About the authors

Ahmed Patel received his MSc and PhD degrees in Computer Science from Trinity College Dublin (TCD) in 1978 and 1984, respectively, specializing in the design, implementation and performance analysis of packet switched networks. He is a Professor in Computer Science at Universiti Kebangsaan Malaysia and a visiting professor at Kingston University in the UK. He has published over 200 technical and scientific papers and co-authored several books. He is currently involved in the R&D of cybercrime investigations and forensic computing, intrusion detection & prevention systems, cloud computing autonomic computing, web search engines, e-commerce and developing a framework and architecture of a comprehensive quality of service facility for networking protocols and advanced services. He is a member of the Editorial Advisory Board of the following international journals: *Computer Standards & Interface*, *Information Management & Computer Security* and *Cyber Criminology*. Ahmed Patel is the corresponding author and can be contacted at: whinchat2010@gmail.com

Wei Qi received her Bachelor of Applied Science in Computer Science from Royal Melbourne Institute of Technology (RMIT) in 2008, and her Master degree in Computer Science from Universiti Kebangsaan Malaysia (UKM) in 2011. She completed an industrial research project on operating system security during her final year of bachelor studies. Her Master's research focused on the design and implementation of framework system architecture for secure and trustworthy mobile agent-based e-marketplace with her supervisor Prof. Dr Ahmed Patel. She is interested in distributed computing, security of mobile agent-based systems, forensic investigation and network security. She has published two papers.



Mona Taghavi, (a.k.a. Malake ye Zibe va Shirin) received her BSc degree in Information Technology from Parand Islamic Azad University of Iran in 2007. Besides her involvement in several Iranian national ICT research projects, she had worked for an IT consulting and project managing company which was responsible for overseeing and preparing some of the technical reports for the Supreme Council of Information and Communication Technology (SCICT) of Iran programme. Currently, she is pursuing her MSc in Information Systems at Universiti Kebangsaan Malaysia and undertaking research in cooperation with Prof. Dr Ahmed Patel in advanced secure web-based information systems and secure mobile agent-based E-marketplace systems. She has published four papers and is a reviewer of papers for *Computer Standards & Interface Journal.*