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

Design of secure and trustworthy mobile agent-based e-marketplace system

Ahmed Patel
School of Computer Science, Faculty of Information Science and Technology, Centre of Software Technology and Management (SOFTEM), Universiti Kebangsaan Malaysia, Bangi, Malaysia and Faculty of Computing Information Systems and Mathematics, Kingston University, London, UK, and
Wei Qi and Mona Taghavi
School of Computer Science, Faculty of Information Science and Technology, Centre of Software Technology and Management (SOFTEM), Universiti Kebangsaan Malaysia, Bangi, Malaysia

Received March 2011
Reviewed May 2011, July 2011, August 2011
Accepted August 2011

Abstract

Purpose – Mobile agent-based e-marketplace is one type of business application that has been developed as a flexible and efficient approach to help companies or corporations to extend their businesses to outreach larger markets without regional and continental boundaries. However, every distributed system is unable to avoid the security problems due to the open internet environment. Mobile agent-based e-marketplaces are no exception. Thus, the security of mobile agents is a crucial factor in the design of mobile agent-based e-marketplaces. To overcome this kind of problem, the purpose of this paper is to design and implement a framework and system of secure and trustworthy mobile agent based e-marketplace.

Design/methodology/approach – This paper presents the system design for the system implementation based on the designed framework. It includes three major aspects: the design issues, system design and development environment and tools for system implementation. The system architecture, use case diagram and use case specifications are presented in the system design section.

Findings – The system design is an essential step that is required before a prototype system is implemented. The system is designed based on the described and outlined requirements and evaluation criteria, therefore, to support a secure and trustworthy trading environment. The paper is concluded by discussing and highlighting further research work.

Originality/value – This paper presents the system design for implementing a secure and trustworthy mobile agent-based e-marketplace system by using the latest version of UML modeling tool and techniques.

Keywords Design, E-commerce, E-marketplace, Mobile agent, Security, Use cases, Tools, Trust

Paper type Research paper

1. Introduction

Mobile agent technology has attracted a lot of interest in recent years. It has been proposed by researchers as a useful technology for developing mobile agent-based e-marketplaces since 1994. Mobile agent provides a number of advantages such as autonomy, flexibility, efficiency and effective usage of bandwidth. All such features apply to the variety of e-marketplace models (Yang, 2005). However, the development

Information Management & Computer Security
Vol. 19 No. 5, 2011
pp. 333-352

0968-5227
DOI 10.1108/09685221111188610

of mobile agent-based e-marketplaces have a slow uptake due to the lack of advanced applications, technologies, safe operating platforms and supporting environments and the confidence to go with it. One of the most valuable characteristics of mobile agents is their mobility that enables them to travel autonomously through the network. This property is precisely the reason that mobile agents are exposed to different types of threats. Strong mobility causes high security risks and threats, while weak mobility causes low security risks and threats (Zhang and Lin, 2005). Therefore, mobile agents raise a number of security issues such as the protection of platform/host that runs the mobile agent against attacks which can harm or use its resources without permission. Also, the issue of protection of mobile agents and their supporting systems against malicious attacks from a variety of intervening sources that might alter information they carry and process it when they visit the hosts in its transactions' itineraries or schedules. For this reason, safety measures should be embedded to ensure buyers and sellers confidence against attacks in the mobile agent-based e-marketplace applications.

Safety measures, particularly 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 Transport Layer Security and its predecessor Secure Socket Layer (SSL) and Secure Electronic Transaction (SET) (Drew, 1999) for internet payment in 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. SET on the other hand is a convenient system for ensuring the security of financial transactions whereby a user is given an electronic wallet in the form of a digital certificate and a transaction is conducted and verified using a combination of digital certificates and digital certificates among the buyer, the seller and the participating bank in a way that ensures privacy and confidentiality. It uses Public Key Infrastructure for privacy and X.509v3 digital certificates to authenticate participants in 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).

Nowadays, when mentioning security, audit and digital forensic techniques appear as major issues (Antoniou *et al.*, 2008; Katos and Patel, 2008) because for normal business transactions audit trails have to be kept and in the event of a suspected criminal activity, digital forensics through investigations and evidence presentation are prerequisite requirements. The rapid inroads made by cyber-criminals and keeping a track and proofing their criminal activities on internet is very high on the agenda of international law enforcement and standards bodies (Cerezo *et al.*, 2007). In the e-marketplace environment, real-time digital forensics framework includes several components such as data collection, data analyzing, protocols, to identify digital forensics, and report the digital evidence, etc. We can use the static and/or mobile management agents to carry out these component functions, therefore, to improve the intelligence, self-adaptability,

flexibility and fault tolerance in the secure and trustworthy mobile agent-based e-marketplace (STMAE) distributed network system. Using mobile agents to represent the digital forensics functions, it can automatically collect the network data from multiple distributed heterogeneous system, thus, it can efficiently reduce the data storage requirements of a single monolithic system, reduce the bandwidth and communication overhead significantly by correlating only pertinent data (Patel, 2005).

Although research works in the literature (Song and Korba, 2003; Zhang and Lin, 2005; Zhao *et al.*, 2007) for designing mobile agent-based e-marketplaces can be found, the lack of standards for 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, which is defined by Jailani *et al.* (2008) and Patel (2010). They investigated eight security concerns of the mobile agent-based marketplaces, namely security, privacy, safety, trust, digital forensic, malicious agent and payment. It also includes the associated protocols for mobile agent to perform e-marketplace activities, keep tracking illegal activities and record the digital evidence for e-marketplace to make an STMAE environment. Therefore, we should take into consideration as regards such issues when we design a secure and trustworthy framework for modeling mobile agent-based e-marketplace. So far, 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 reviewed literature, we have investigated the requirements and evaluation criteria for designing and implementing our proposed framework and system including infrastructure services, safety measure services and technical supports for system implementations, in which the safety measure services investigated security concerns such as the cryptography and safety measure protocols which include the secure migration protocol, secure payment protocol and digital forensic protocol (Qi and Patel, 2009). Thus, we proposed a framework of STMAE including the safety measure services (cryptography and safety measure protocols) to solve the security problems (Qi and Patel, 2009). Figure 1 shows the architecture of the proposed framework of STMAE.

The goal of this paper is to present the system design of STMAE for the system implementation. The development environment and tools for the system are also discussed in this paper since it is part of system design for further implementation. The paper is organized as follows. Section 2 represents the design issues from system design level and implementation design level which includes the safety measure issues, design tools issues, and programming platform and language issues. Section 3 discusses the system design which includes the system architecture, the use case diagram and use case specifications. Section 4 describes the development environment and corresponding tools for further system implementation. Finally, we discuss and conclude further research work in this subject area in Section 5.

2. Design issues

The design issues are presented from two levels, the system design level and implementation design level. The system design level includes safety measure issues and design tools issues for system design. The implementation design level includes the programming platform and language issues.

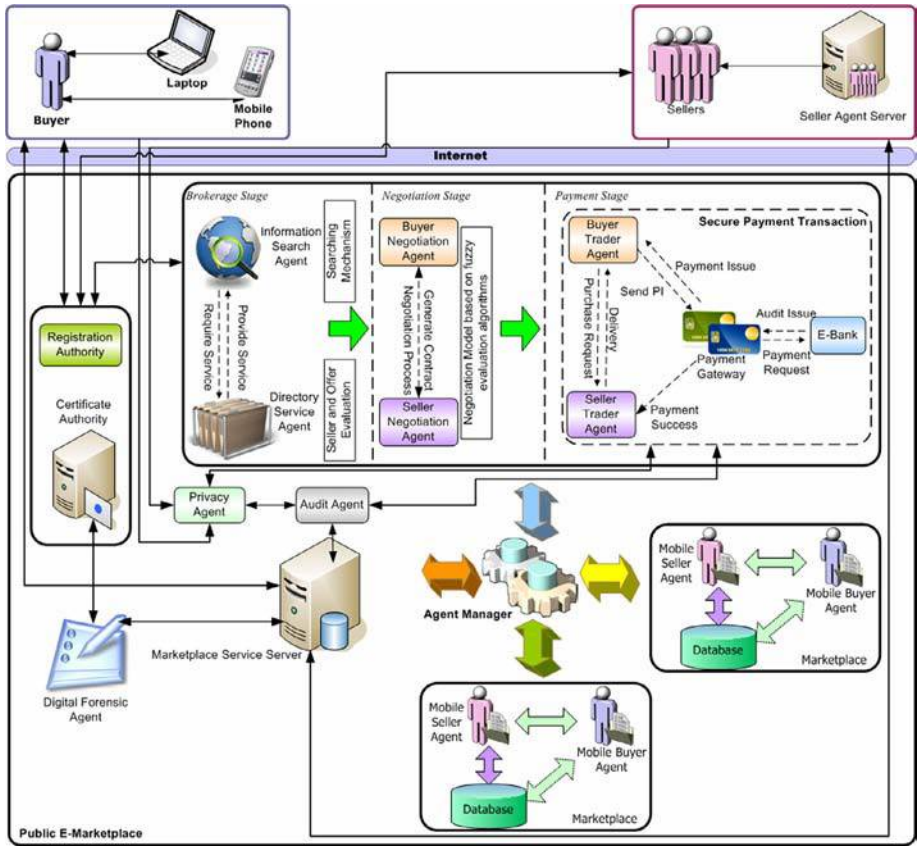

Figure 1.
Conceptual framework of STMAE

2.1 Safety measure issues

There is a set of safety measures defined by Patel (2010) which include eight security concerns and requirements of mobile agent-based e-marketplaces. These issues, together with the associated challenges previously discussed, are the basis for establishing the security requirements to integrate into an STMAE framework. These concerns are described as follows:

(1) *Security.* Mobile agents always cause security issues in the e-commerce applications. These security issues involve authentication, authorization, integrity, confidentiality, encryption, and other operational defense facilities. The security threats can be classified into four categories, as described in our literature paper (Patel *et al.*, 2010). The common threats are masquerading, denial of service, unauthorized access, annoyance attack, eavesdropping, alteration and repudiation. These threats cause different levels of damage in the e-marketplace environment. Therefore, the e-marketplace should offer a set of security mechanisms to prevent these threats. Generally, mobile agents can use either single hop or multi-hop for their mobility characteristic in order to support an initial level of security. To make the mobile agent environment more secure,

Venkatesan *et al.* (2010) proposed an advanced model for mobile agents to improve upon the efficiency and security levels of the existing Malicious Identification Police model for scanning the incoming agents to detect malicious activities and to overcome vulnerabilities in the existing Root Canal algorithm, which performs code integrity checks.

(2) *Privacy.* Privacy is an expression of self-determination and dignity perceived as a fundamental human right by most constitutions of democratic societies. Personal privacy is primarily based on data protection legislations and directives to protect, collect, store and process personal data in order to guarantee privacy. Both legal and technical are required to protect privacy and ensure that the individual or the mobile agent systems (MASs) performing on behalf of the individual has explicit control to guarantee that right to protect privacy. Privacy enhancing technologies (PETs) is a system of ICT measures consisting of a set of computer tools, applications and mechanisms protecting privacy related information by eliminating or minimising personal data by allowing online users in e-marketplace applications to protect the privacy of their personally identifiable information by such services or applications without the loss of the functionality of the information system, thereby preventing unnecessary or unwanted processing of personal data (van Blarkom *et al.*, 2003) to mainly overcome privacy invasive technologies (PITs) (Patel, 2010). To protect against PITs, PETs should ideally be interlaced and interoperated into the trace, audit and digital forensic investigation subsystem/components in order to detect malicious attacks and intrusions. PETs may or may not use encryption depending on the offered options for privacy protection. For example, encryption might be used to protect e-mail, documents and transaction exchanges from being read by other parties. To address user concerns regarding privacy and trust issues in a mobile agent-based e-marketplaces, Au *et al.* (2004) introduced the concept of establishing trust by making use of referrals from an external third party in the form of anonymous attribute certificates to ensure privacy. The key point is that the STMAE system embodies the concept of privacy by design to protect users.

(3) *Safety.* The safety of the system should provide reliable backup systems with appropriate technical management support functions. These are used to ensure that all the transactions performed within sessions using multiple integrated resources are securely operated and are properly recorded and not lost in the event of system failure. Wang *et al.* (2007) proposed a migration mechanism of security which included the trust services and fault tolerance for mobile agents to overcome the consequences of such failures. In their work, the trust service is duplicated as backup at several different trusted nodes in the e-marketplace environment. When the trust services expire, the system can select one backup from the trusted nodes using an election algorithm scheme (Krzyzanowski, 2000). However, this mechanism has some drawbacks such as the expiry of trust services and trusted nodes that affect the backup and recovery procedures which may hamper the ability of mobile agents to operate and travel within time constraints due to transmission problems.

(4) *Trust.* The e-marketplace should provide the trust environment for buyers and suppliers. Trust services are used to verify whether a mobile agent is a legal agent

or otherwise through authorization and authentication processes in order to prevent the operation of malicious agents masquerading freely in e-marketplaces. A trusted third party (TTP) may be needed as a certificate authority for the management and maintenance of keys for encryption and other identification and verifications purposes. Warnier *et al.* (2009) proposed a secure migration protocol for mobile agents based on the distribution of trust to ensure that breach of integrity in migration paths of mobile agents in large-scale distributed agent systems would be detected. This approach distributes trust over three hosts during each migration step. The combination of sequence numbers with signatures, guarantees that one or more hosts can detect whether part of the migration path, including cycles, has been removed. The approach works well in situations with only one malicious host in a migration path, or in environments with multiple malicious hosts that do not conspire together. However, if multiple malicious hosts conspire together the situation becomes more complex.

(5) *Digital forensic.* Digital forensics is an activity of investigation to trace and analyze illegal and fraudulent events to produce evidence for the purpose of law enforcement (Patel, 2010). It includes the event data collection, analysis of the event, the employment of the forensic protocol to detect the illegal activities, to report cyber crimes and record the evidence for law enforcement purposes. Antoniou *et al.* (2008) proposed a security protocol called ERPINA protocol for privacy and forensics investigation purposes which increases the level of the protocol's dual purposes and reliability.

(6) *Auditing.* Auditing can be a part of digital forensics or vice versa, in a mobile agent management framework. The e-marketplace should be able to audit the legality of all the participants' activities and events based on the proper accounting, security and privacy policies. Patel (2010) and Jailani *et al.* (2008) describe the digital forensics investigation and audit scenario for the mobile agent-based e-marketplaces. The scenario portrays a malicious anonymous user attempting to abuse the e-marketplace environment. The auditing function has the ability to detect attacks in such an environment during normal management accounting exercises as a legal requirement.

(7) *Confidentiality.* The e-marketplace should only permit authorized parties to operate in order to ensure that all the participants in the marketplace are legal entities who maintain each other's confidentiality. The mobile agents only work for these authorized entities that either carry the credentials and information to perform the tasks or store the data on a trusted platform entities. Liu *et al.* (2010) proposed an anonymous authentication scheme for mobile communication to ensure the confidentiality of the user's identity, whereabouts and other information. This scheme is based on binary tree theory and zero knowledge proof, and combines it with divisible e-cash theory that implements an encrypted transmission mechanism for mobile agents which overcomes the weakness of traditional communications for the distribution of a data encryption key without going through a third-party to ensure better confidentiality.

(8) *Payment.* The payment process should provide the secure auditable mechanisms for automated electronic payment transactions. A secure payment protocol should be provided by e-marketplaces for the participant to facilitate the payment

process in a secure environment through mobile agents. In recent literature, Ou and Ou (2010) proposed a secure payment protocol named Agent-based SET with Non-repudiation (SETNR/A) protocol to improve the weakness lacking for non-repudiation mechanisms from the SET and SET/A protocols for credit card-based transactions. One major advantage of this agent-based payment protocol is to reduce inconvenience for mobile clients such as connection time and search for suitable merchant servers with the ability to provide security during mobile payment transactions.

2.2 Design tools issues

There are number of reasons why Unified Modeling Language (UML) was chosen as the notation for the system design. The UML (OMG, 1997) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development (Medvidovic *et al.*, 2002). It combines the best techniques from data modeling, business modeling, object modeling and component modeling. It can be used in different processes, not only throughout the software development life cycle but also across different implementation technologies (Mishra, 1997). The purpose of the UML is to be a standard modeling language which can model concurrent and distributed systems. However, it is not a development method by itself which was designed to be compatible with the leading object-oriented software development methods of its time (Hunt, 2000). Since the evolution of the UML, some of these methods have been recast to take advantage of the new notations and new methods which have been created based on UML.

Nowadays, with rapid software development, there are over hundred UML modeling tools in the world. To choose a suitable UML modeling tool, we classify ten popular UML modeling tools which list eight commonly used chart types (also known as different types of diagrams) and its code generation to make a comparison between these tools. The comparison also covers the platform/OS where these tools implement and some comments of these tools whether they are suitable for our modeling or not. The XMI factor is also considered since it is the interchange format for UML models to be exchanged among the UML tools. Figure 2 shows this comparison between these different UML modeling tools.

Based on the comparison results shown in Figure 2, we select Jude Community as our UML modeling tool according to several advantages it owns. Jude Community version 5.5.2 (Change Vision Inc., 2009) is a free UML modeling tool created by the Japanese company Change Vision Inc. that facilitates object-oriented design of software by using Java and UML. The modeling features of Jude Community have been designed to be simple, efficient and have a flexible modeling construction and with user-friendly interfaces.

2.3 Programming platform and language issues

Numerous systems were developed based on mobile agent software technology in recent years and have proved to be very useful in actual service deployment. Even now some of the systems are still currently under development. These MASs are significantly different and not interoperable although all of them share the same Java-enabled basic characteristics such as mobility, network class loading on demand and communication mechanisms based on message passing. In order to identify both the commonalities and

Tools Name	Creator	Open Source	Supported Diagram Types								Code Generation				XMI	Platform/ OS	Comments
			Class Diagram	Use Case Diagram	Sequence Diagram	Activity Diagram	State Chart Diagram	Collaboration Diagram	Component Diagram	Deployment Diagram	CORBA-IDL	C++	Java	Others			
ArgoUML	Tigris.org	Yes	✓	✓	✓	✓	✓	✓	✓	✓	⊘	⊘	✓	—	✓	Java (Cross-Platform)	Easy to use, but very slow
Dia	Alexander Larsson	Yes	✓	✓	✓	⊘	⊘	⊘	⊘	✓	✓	✓	✓	—	⊘	Linux, Irix 6.5	Easy to use, but no semantics
Eclipse UML	Eclipse Foundation	Yes	✓	✓	✓	✓	✓	⊘	✓	✓	⊘	⊘	✓	—	✓	Java (Cross-Platform)	Eclipse-Integration
Frame UML	Frame	Yes	✓	✓	✓	✓	✓	✓	✓	✓	⊘	✓	✓	Java Script	✓	Windows	Java (Partial), Support UML 2.x.x
Jude Community	Change Vision Inc.	Yes	✓	✓	✓	✓	✓	✓	✓	✓	⊘	⊘	✓	—	⊘	Java (Cross-Platform)	Easy to use, Efficient, Flexible
NetBeans UML Plugin	NetBeans	Yes	✓	✓	✓	✓	✓	⊘	⊘	⊘	⊘	⊘	✓	—	⊘	Java (Cross-Platform)	NetBeans 6.01DE 5.5 Plugin
StarUML	Plastic Software	Yes	✓	✓	✓	✓	✓	⊘	⊘	⊘	⊘	✓	⊘	Delphi C#	⊘	Windows	Efficient, Flexible, Expandable
UMLet	Martin, Thomas	Yes	✓	✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	—	✓	Windows Linux, Eclipse	Efficient, Eclipse-Integration
Visio	Microsoft	No	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	C# VB	✓	Windows	Simple, but not efficient & flexible
Visual Paradigm for UML	Visual Paradigm Int'l Ltd.	No	✓	✓	✓	✓	✓	✓	✓	✓	⊘	⊘	✓	—	⊘	Java (Cross-Platform)	Easy to use, Eclipse-Integration

Symbol: ✓ Yes ✓ Partial ⊘ No

Figure 2.
Comparisons between different types of UML modeling tools

the differences among Java-based MASs, several MASs were selected according to some criteria that the platforms were widely cited, robust and stable, very representative of on-going research and commercial efforts, easily documented and extensively used for developing a broad range of applications and sophisticated programming systems. In particular, Aglet (IBM, 2004), Grasshopper (IKV++ GmbH, 2003), Voyager (ObjectSpace Inc., 2003) and Ajanta (Ajanta, 2003) are considered.

Several essential issues should be taken into consideration for the comparison of these mobile agent frameworks. Fortino *et al.* (2008) identified the different issues for comparing those mobile agent frameworks mentioned above 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 done by Jha (2002) includes qualitative and quantitative comparison for e-commerce applications across three Java-based mobile agent frameworks namely Voyager, Aglet and Concordia. According to their research works, agent services available in Aglets for a 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. Therefore, IBM Aglet is selected as the mobile agent platform for the system implementation. The features of Aglet such as mobility, autonomy, rapid response time, concurrency and local interaction were satisfactory

for our e-marketplace. Aglet can run on any machine that supports the Aglet API. It runs on Tahiti server and may adopt seven different states during their lifecycle.

3. System design

Since the system design is the main focus of this paper, we present the system architecture and use cases with its corresponding use case specifications in the following subsection.

3.1 System architecture

The system architecture is designed based on the conceptual framework of STMAE. It is used for mapping of components in the conceptual framework for system implementation. We apply mobile agent technology with safety measure services in the client/server model for the system design and implementation. The system is implemented based on IBM Aglet (IBM, 2004) by using Java programming language. By using Eclipse (Eclipse.Org, 2010) programming software to implement the system so that it can easily be integrated with other cross-middleware systems implemented on the same platform. The Aglet server is presented as a mobile agent server which is used to identify components that run as applications and what these system components are called in the conceptual framework. Figure 3 shows the system architecture for the STMAE.

According to Figure 3, the system architecture consists of at least four Aglet servers known as the marketplace service server (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 ASs 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 server with system components and its 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 buyer and seller. It is responsible for maintaining the information for a set of marketplaces. To start up this server, it should register at the AS to get the authorization for service providing.

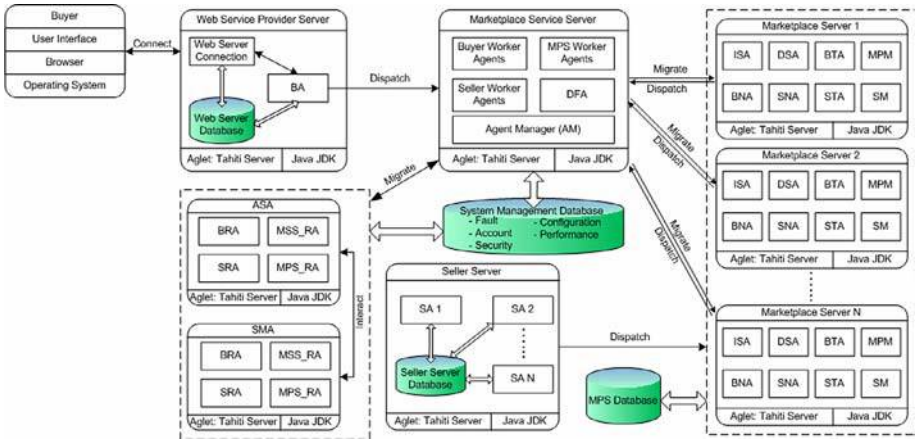

Figure 3.
System architecture of STMAE

It is used to provide the services for mobile agents from both buyer and seller sides to perform the tasks such as searching the information of the specified product and providing the offers of the products. There are several mobile agents 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 presentation of a set of mobile agents that work for 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 present the RA of MPS (MPS_RA), the RA of Seller (SRA) and directory service agent (DSA). For those agents that run on this server, they can be divided into stationary agents and mobile agents. AM is responsible for managing the work flow of mobile agents in the public e-marketplace. It can be viewed as the stationary agent, while other agents that run on this MSS are all recognized as mobile agents. The mobile agents can migrate to different Aglet server 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 investigation as evidence.

Authority server. AS acts as a TTP in the public e-marketplace. It runs two different ASs, 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 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 buyer agent (BA), seller agent (SA), MSS and MPS. In addition, it is responsible for recording the evidence investigations and auditing on participants in the e-marketplace according to the investigation reports. An authentication database which contains security attributes (such as certificate, etc.) is used for the authentication and verification. This database also records the successful authentication and verification of mobile agents.

Marketplace server. MPS consists of a set of marketplaces and SAs 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 SAs. It also has the ability to authenticate incoming foreign mobile agents. This server is an execution environment for the incoming mobile agents such as ISA, BNA, BTA, DSA, seller negotiation agent (SNA) and seller trader agent (STA). It has the ability to 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 manger (SM) are the two stationary agents, while all other agents are recognized as the 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. It can authenticate incoming foreign mobile agents and monitor the communication out of the marketplace from BAs or SAs, and broadcast the sellers' certificates 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 SA and cancellation of trading transaction. It is also responsible for managing and maintaining the directory of SAs in the DS.
- *DS*. It is responsible for DSAs running on the server. The DSA maintains the seller's and product's information which were recorded in the MPS database. DSA will periodically send the updated information to MPM for modifying the product's catalogue of the SAs that are maintained in the DS.

Seller server (SS). It is responsible for a number of SAs running on the server. The SAs that run on this server are in charge of dispatching worker agents to communicate with incoming BAs, provide the requested product information, monitor the execution of BAs and protect the local resources of the SAs. At the same time, register the SAs to the MPM and through it register to the MSS when the seller is set up. It applies the certificate of the SA from SMA and sends it to the MPM. An SS 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. Buyer should register at this WSPS for web service. The WSPS allows 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 BA has the ability to create slave mobile agents at WSPS and distribute tasks to these slave mobile agents. This master BA is the stationary agent running at this server. All the buyer worker agents are dispatched from this server.

3.2 Use case diagram and use case specifications

The basic use case diagram and use case specifications are two of the most important elements for system design. The use case model introduces a consistent approach for the mobile agent-based functionality of STMAE. Figure 4 shows the main use case diagram for STMAE. The use case specifications can be represented with several components such as the actors of the specific use case, the description and scenarios of the use case, the pre-condition and the post-condition to fulfil the use case and the error with the error handling if the failure of tasks performing occurs. According to Figure 4, the use case specifications for each use case are described as follows.

Use case 1: create mobile agent. This use case allows both buyer and seller to create their own mobile agents on behalf of them to perform specific tasks in the e-marketplace. They can create the master mobile agent or slave mobile agents. User (buyer or seller) creates a master agent at local host. The master agent has the ability to create slave mobile agents and distribute tasks to these slave mobile agents. They can dispatch these slave mobile agents to access into the marketplace to perform the specific tasks. The pre-condition of this use case is that the buyer/seller wants to purchase/sell products/services and users should have ability to employ mobile agents to work for them. Obviously, the post-condition is that users (buyer or seller) successfully create the master agent and slave mobile agents are created by the master agent if they were needed. However, if the creation of the mobile agent fails, the system will either send the error message to the user or to the master mobile agent as the error handling.

Use case 2: configure mobile agent. In this use case, mobile agents should have compatibilities to configure the specific business process or transactions in the e-marketplace environment. It is possible to configure mobile agent not only for single tasks,

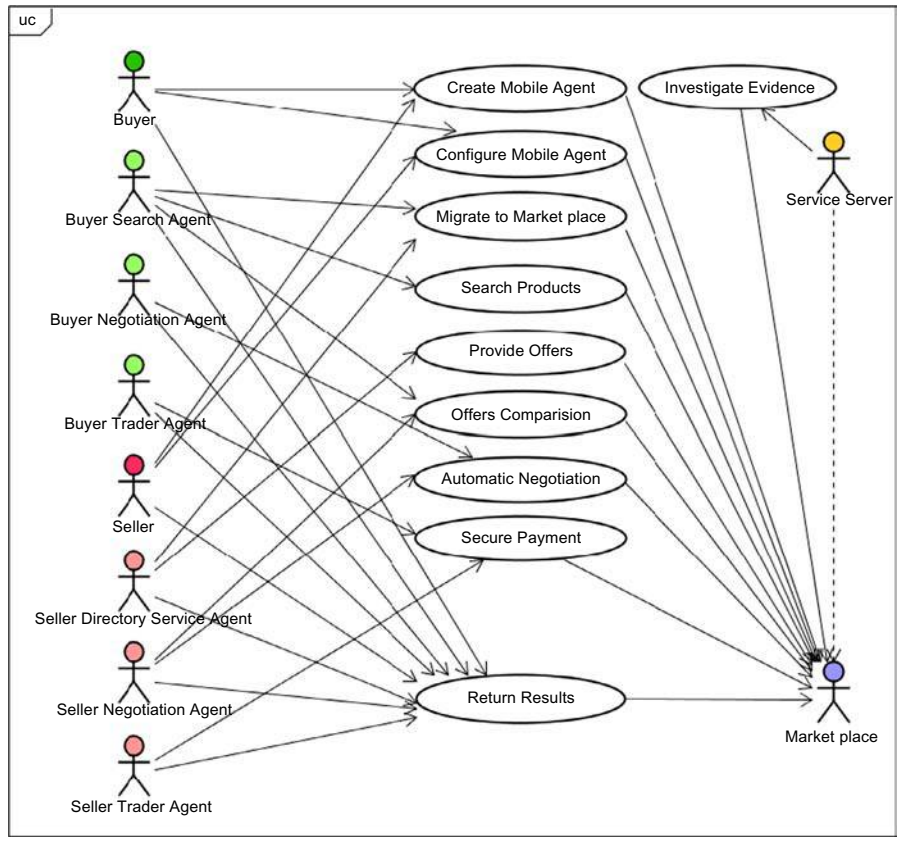

Figure 4.
Main use case diagram of STMAE

but also for a whole task flow. The configuration of mobile agents is in a transparent way since different scenarios follow the same communication protocol. Several scenarios here can be represented. First, the user configures a mobile agent through the service server either for the remote work at the e-marketplace which involves multiple marketplaces. The user sends an HTTP request including the task data and the migration route to the service server. The service server interacts with the mobile agent and sends the specific message including the configuration parameters to the mobile agent. Second, the user configures the specific mobile agent from its local host for performing tasks on multiple marketplaces. Thus, the client application creates a mobile agent and distributes data including the configuration settings and the route information to that mobile agent. Third, the master mobile agent configures a slave mobile agent for performing local work. Therefore, the master mobile agent communicates with the slave mobile agent and sends the configuration data via messages. The mobile agent is now successfully configured and is able to start proceeding. Thus, the mobile agent to be configured must be alive and idle as the pre-condition of this use case. Otherwise, there is configuration failure and the mobile agent should send the error message to the master agent or user.

Use case 3: migrate to marketplace. Most of the mobile agents should migrate to the marketplace except the stationary mobile agent on the home agent server. The actors from buyer side are the BA, ISA, BNA and BTA. The actors from seller side are the SA, DSA, SNA and STA. This use case allows mobile agents access into the e-marketplace, initiate and perform tasks, and finally return results to the end-user. It enables the e-marketplace to offer two features: migrate to the marketplace and interconnect to multiple marketplaces. In a general case, the mobile agents carry the retrieved data from the master mobile agent and migrate to the marketplace through a secure migration protocol (Qi and Patel, 2009). The mobile agent should be authenticated and verified through the TTP when it migrates to the marketplace. The mobile agents have the ability to migrate from the service server to another marketplace according to its route settings. Last, mobile agents migrate from the service server back to the home server. However, if network connection failure happens, the mobile agent tries to migrate again or select another destination for migrating. If both are not possible, the mobile agent sends an error message back to the user if it works on the home server, or it waits until the home server is available again.

Use case 4: search products. This use case allows buyer to search the specific product/service from the e-marketplace by employing the ISA. The master mobile agent distributes the search task to ISA. This search agent carries the task including the corresponding product information and access to the e-marketplace. The service server queries the products from the database while the search agent waits for the results. When the product querying is completed, the service server sends the search results to the search agent. This search agent receives and stores the search results. If an error has occurred due to connection failure to the services server or the search data is incorrect, then, it either can re-connect to the service server or correct the incorrect search data.

Use case 5: provide offers. This use case allows DSA to provide offers in the e-marketplace for buyer search agent to retrieve the target offers from the service server. The DSA carries the offer list from the seller master agent. It migrates to the service server and the service server records the offer list for further search operations. If offers are successfully recorded in the service server for searching, the DSA sends the message back to the master mobile agent and these offers' are valid for searching. Otherwise, if the offer list fails to record in the service server, then, the DSA reports the error to the user.

Use case 6: offers comparison. In this use case, the buyer search agent can compare multiple offers according to the user's preference for comparison. This user's preference includes several aspects such as product specification with its brand, model, price, delivery time, warranty, and the reputation and the security level of sellers. The buyer search agent carries these criteria from the buyer and migrates to the marketplace for the task performing. When it arrives at the MSS, it starts to compare the offers through the offer comparison mechanism. If the buyer search agent has the corrected comparison criteria, it should store the comparison results. However, if the comparison criteria were not corrected, the comparison process would not be completed. The buyer search agent should store the error message and report it to buyer.

Use case 7: automatic negotiation. This use case allows the BNA and SNA enter to the marketplace to perform the automatic negotiation process via the service server. It should be absolutely transparent what actions have been done at the negotiation process through the proposed negotiation protocol (Qi and Patel, 2009). Additionally, negotiation agents can be authorized by either buyer or seller users in activating

the negotiation process to reach a deal. The server stores the offered pre-contracts in every step as a confidant for both trading partners in the database. Initially, the master mobile agent should distribute the negotiation task to the negotiation agent. Following, the service server receives the negotiation requests from both sides and the negotiation process starts. The negotiation can be terminated when SNA and BNA achieve their desired price. However, it may include some other situations such as when the negotiation meets the round limited as buyer's preference or the proposed counter offer reaches the border offers that are offered by sellers. The negotiation results are stored when the negotiation is successful. Otherwise, the negotiation process fails due to the connection failure or incorrect data the negotiation agent carries. Then, the negotiation agents store the error message and return it to both sides.

Use case 8: secure payment. This use case allows trader agents from both sides (buyer and seller) to perform the secure payment through the payment gateway which follow a secure payment protocol (Qi, 2011). It starts after the automatic negotiation, the trader agent from buyer side confirms to purchase the product from the target seller. Buyer completes the payment information, the trader agent then carries this authenticated information to the TTP for the payment request. The payment process is then initiated. The trader agent from the seller side receives the purchase request from buyer, and the payment issues from the TTP. When the payment is confirmed from a seller side, the STA sends the confirmation to both TTP and BTA. To make this payment process successful, the buyer first should confirm that the user wishes to purchase the corresponding products and then, the system requires both buyer and seller to have valid credit cards. If the credit card has expired or lack enough credit, the payment transactions fail. Trader agents from both sides store the error message and the payment is terminated.

Use case 9: investigate evidence. This use case allows the service server to capture the packet from the different nodes when they access the service server. The main actor of this use case is digital forensic agent which acts as the real time recorder that investigates and records the digital evidence following a digital forensic protocol (Qi, 2011). All the evidence will be recorded in a log file and listed in the forensic table on the results interface. When service server starts working, the digital forensic is enabled. If the digital forensic agent is not working properly, the log file cannot be created and records evidence failure. Thus, it should send an error message to the service server.

Use case 10: return results. Every mobile agent should be an actor in this use case. From buyer side, it should include the BA, ISA, BNA and BTA. From seller side, it should include the SA, DSA, SNA and STA. This use case describes the process of mobile agent, which sends back results after tasks have been fulfiled. It is possible to return multiple results back to the specific actor in parallel, if this mobile agent configured to the marketplace for the multiple tasks performing. There are two scenarios when mobile agents return the results:

(1) The specific task performing mobile agent sends the results back to the user via the service sever. After receiving the confirmation message from master mobile agent, this specific mobile agent is disposed.

(2) The mobile agent returns the results by sending a message back to the user on its home agent server. The mobile agent is disposed after user receives the results.

However, this transaction may fail if the network connection fails. Thus, the system will try to re-connect the network and send the message again.

4. System development environment and tools for system implementation

The system development environment and tools should be defined for further system implementation since they are also part of system design. There have several requirements for the software and hardware for implementing the mobile agent-based e-marketplace. For the software, basically, the Windows operating systems are suitable for the software development. Windows XP operating system is selected for the development environment. For the hardware, the developer can either choose laptops or workstations. Furthermore, the computer should be able to connect to the internet for both implementation and testing. The system is implemented in Java-based technologies applicable to mobile agent system: Aglets as the system platform and Java for web-based system to realise prototype e-marketplace system with Extensible Markup Language (XML) and some security components. As one of the goals of the work, the system should implement in such a manner, 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 server platforms. The development tools (such as Aglet, XML, Eclipse, etc.) are described below:

- *Aglet* (IBM, 2004). The marketplace system is implemented by using a mobile agent-based platform Aglets mobile agent Software Development Kit version 2.0.2 which runs on Tahiti Server. To start up with the Tahiti Server, the developer should type the corresponding commands "C: \aglet\bin > agletsd –f../cnf/aglets/props –port 4000", and the Aglet viewer appears. Then, the mobile agents can be created on the Aglet server to perform the functions.
- *Apache Tomcat* (Bidgoli, 2010). It 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. JDK 1.5 was selected as the platform because it is widely available on a number of server platforms.
- *Eclipse* (Eclipse.Org, 2010). It is a platform which interoperates technology that has been designed from the ground up for building integrated web and application development tooling. The advantages of Eclipse are reusable, trustworthy, confidentiality, quality, clarity, longevity and flexibility. Eclipse provides a common user interface model for work developing with tools.
- *XML* (W3.Org, 2003). It describes structure of data and focuses on its semantics. Thus, the structured documents could be exchanged by using XML. It can describe components since it allows a meaningful semantic description. An XML description can be converted to Java document model object so that it can be merged into the service registry system. Normally, mobile agents are subject to strong security restrictions, which are enforced by the security manager or authenticated and verified from the security authority, so mobile agents should find the corresponding services that help complete their tasks. XML is used to describe both the service description and mobile agent's queries.

Thus, XML enables exchange of information not only between different computer systems but also across language boundaries.

- *Jpcap Library* (Fujii, 2007). It is a Java class package that allows Java applications to capture and/or send packets to the network (Fujii, 2007). Jpcap is developed based on libcap/wincap and raw socket API. Therefore, Jpcap is supposed to work on any operating system on which libcap/wincap has been implemented. It supports the following types of packets: the ethernet, IPv4, IPv6, ARP/RARP, TCP, UDP and ICMPv4. It also supports sending raw packets.

As discussed above, the system development environment and tools for our system implementation are selected based on the current techniques and technologies. They may change or upgrade in future if new techniques and technologies appear or the new version of the selected software is released.

5. Conclusion

This paper described the system design for the system implementation based on the proposed conceptual framework (Qi, 2011). The design approach is using the UML software engineering methodology. The system design is presented from system architecture and uses cases with its specifications. The system architecture is used to map the components from the conceptual framework. The functional processes are presented in use cases diagram and use case specifications. These use cases with safety measure services were defined based on the requirements and evaluation criteria, which we had described and outlined (Qi and Patel, 2009; Qi, 2011). The system development environment and tools were also presented since it is part of the system design for the system implementation. It includes the software and hardware for the system, and essential development tools such as Aglet, Eclipse, Apache Tomcat, XML and the Jpcap Library.

For further research of this subject area, it would be implementing and testing a system prototype of proposed STMAE based on the system design, we have presented in this paper. A prototype was implemented in order to analyze, evaluate, verify and validate the proposed conceptual framework. It was implemented in Java-based technologies, applicable to the mobile agent system: Aglets as the system platform and Java EE for web-based system to realise a prototype e-marketplace system with XML and security components. However, it would require 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 sellers, detect and record 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 future research. We have evaluated the solution against the requirements and evaluation criteria based on the analyzed results from experimental runs, observations, datasets and outcomes against the criteria determined during the requirements capture stage of the research work. The evaluation of solution, which is beyond the scope of the current design paper, is presented in another follow-up paper in this journal. The evaluation results show that the proposed STMAE framework and system have the ability to provide a secure and efficient e-marketplace environment

for trading products, and meet the security, performance, scalability, reliability, portability, modularity requirements for developing mobile agent-based systems with the use of existing technology. The evaluation and performance results are extensively discussed in the follow-up paper in its entirety.

Currently, we have expanded the scope of the research work as follows:

- enhance the user interface with intelligent features that can be used in different mobile devices such as PDA and new generation mobile handsets;
- make the implementation that can be automatically reusable for further system development;
- experiment with different cryptography algorithms in order to measure the performance and benefits and to show that they fulfil 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 protocol to make the payment be more 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; and
- change or upgrade the developing tools for system development to enhance the usability of the system if new techniques appear in future.

We envisage (hope) further research can shed more light and help improve the worthiness of our proposed framework and system in real operational environments.

References

Ajanta (2003), "The Ajanta mobile agent system", Documentation and software. available at: www.cs.umn.edu/Ajanta/ (accessed 22 February 2011).

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.

Au, R., Vasanta, H., Choo, K.R. and Looi, M. (2004), "A user-centric anonymous authorisation framework in e-commerce environment in Janssen", in Sol, M.H.J. and Wangenaar, R.W. (Eds), *sixth International Conference in Electronic Commerce*, pp. 138-47.

Bidgoli, H. (2010), *The Handbook of Technology Management, Supply Chain Management, Marketing and Advertising, and Global Management*, Vol. 2, Wiley, New York, NY, 931 pp.

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)*, pp. 13-27.

Change Vision Inc. (2009), *JUDE/Professional – Design & Modeling UML, ERD, DFD, Flowchart, CRUD, Mind Map*, available at: http://jude.change-vision.com/jude-web/index.html (accessed 5 March 2009).

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

Eclipse.Org (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 23 November 2008).

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.

Fujii, K. (2007), "Jcap – a Java library for capturing and sending network packets", Release Version 0.7, available at: http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html (accessed 19 May 2009).

Hunt, J. (2000), *The Unified Process for Practitioners: Object-oriented Design, UML and Java*, Springer, London.

IBM (2004), "IBM aglet workbench", available at: www.trl.ibm.co.jp/aglets/ (accessed 12 August 2008).

IKV++ GmbH (2003), "Grasshopper mobile agent system: documentation and software", available at: www.grasshopper.de/ (accessed 19 April 2009).

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).

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.

Krzyzanowski, P. (2000), *Process Synchronization and Election Algorithms*, Lectures Note on Distributed Systems, available at: www.cs.rutgers.edu/~pxk/rutgers/notes/content/06-mutex.pdf (accessed 31 December 2010).

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

Liu, J.Y., Gu, L.Z., Luo, S.S. and Yang, Y.X. (2010), "An anonymous authentication scheme for mobile communication", *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS), Beijing, China, 25-27 June*, pp. 359-64.

Medvidovic, N., Rosenblum, D.S., Redmiles, D.F. and Robbins, J.E. (2002), "Modeling software architectures in the unified modeling language", *ACM Transactions on Software Engineering and Methodology*, Vol. 11 No. 1, pp. 2-57.

Mishra, S. (1997), "Visual modeling & unified modeling language (UML): introduction to UML", Rational Software Corporation. available at: www2.informatik.hu-berlin.de/~hs/Lehre/2004-WS_SWQS/20050107_Ex_UML.ppt (accessed 22 February 2011).

ObjectSapce Inc. (2003), "Voyager: documentation and software", available at: www.recursionsw.com/products/voyager (accessed 2 April 2009).

OMG (1997), Introduction to OMG's Unifiied Modeling Language™ (UML®), Object Management Group, Inc., available at: www.omg.org/gettingstarted/what_is_uml.htm (accessed 5 May 2009).

Ou, C.M. and Ou, C.R. (2010), "SETNR/A: an agent-based secure payment protocol for mobile commerce", *International Journal of Intelligent Information and Database Systems*, Vol. 4 No. 3, pp. 212-26.

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, Hershey, PA, pp. 252-64.

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.

Poggi, A., Tomaiuolo, M. and Vitaglione, G. (2003), "Security and trust in agent-oriented middleware", in Meersman, R. and Tari, Z. (Eds), *Proceedings of the OTM Workshops 2003, Catania, Sicily, Italy*, Lecture Notes in Computer Science, Vol. 2889, Springer, Heidelberg, 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, Malaysia.

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.

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

van Blarkom, G.W., Borking, J.J. and Olk, J.G.E. (2003), "PET", *Handbook of Privacy and Privacy-enhancing Technologies* (*The Case of Intelligent Software Agents*), College bescherming persoonsgegevens, The Hague, 372 pp.

Venkatesan, S., Chellappan, C., Vengattaraman, T., Dhavachelva, P. and Vaish, A. (2010), "Advanced mobile agent security models for code integrity and malicious availability check", *Journal of Network and Computer Application*, Vol. 33 No. 6, pp. 661-71.

Wang, Y., Wang, Z.Q. and Wei, L.F. (2007), "A migration mechanism of mobile agent system supporting security and fault-tolerance", *Computer Technology and Development*, Vol. 17 No. 3, pp. 169-75.

Warnier, M., Oey, M.A., Timmer, R.J., Overeinder, B.J. and Brazier, F.M.T. (2009), "Enforcing integrity of agent migration paths by distribution of trust", *International Journal of Intelligent Information and Database Systems*, Vol. 3 No. 4, pp. 382-96.

W3.Org (2003), Extensible Markup Language (XML), available at: www.w3.org/XML/ (accessed 25 May 2009).

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.

Further reading

Wikipedia.Org (2010), UML Tool, available at: http://en.wikipedia.org/wiki/UML_tool (accessed 12 January 2010).

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. He is 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 and 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's degree in Computer Science from Universiti Kebangsaan Malaysia (UKM) in 2011. She did an industrial research project on operating system security during her final year of the bachelor. 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 Professor 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 Mona Taghavi, 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 Professor Dr Ahmed Patel in advanced secure Web-based information systems and Secure Mobile Agent-based E-Marketplace Systems. She has published four papers. She is a reviewer of papers for *Computer Standards & Interface Journal*.