ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 12, December 2013

Implementation of Mobile Agent Architecture for E- Commerce Application for Mobile Devices

Kalpana N. Meher, Prof. Sanjay Jadhav, Prof. P. S. Lokhande

***Abstract* —“Mobility” is a term which newly acclaimed with the increased used on mobile devices. To support business application on remote places this proposed architecture helped by a mobile intelligent agent. By using mobile devices, buyers or sellers can create a mobile intelligent agent; the ability of mobile migration (mobility) on internet agents can do the business on behalf of buyers & seller. Mobile agents are beneficial to do the business anytime and anywhere. Searching for appropriate trader and negotiating with the trader is handled by mobile agent which reduce the time consuming task of the user. As we know that mobile devices owns the limited bandwidth connections with greater expenses. The proposed system architecture can address the problem of expensive connection with limited bandwidth. After launching mobile intelligent agent through network, seller/buyer can disconnect from mobile devices and later on can reconnect to devices for the results. For moving across the network; FIPA standardization provides flexibility & increase interoperability among other agent systems and scalability is provided by proposing system architecture.**

As per market survey; with compared to other mobile operating system, Android captured 80% market share worldwide for that reason the proposed system is designed for Android mobile devices.

For delegating of human users (Buyers / Sellers) on the distributed system environment agents are created and by the migration process on the internet the agents can handle the shopping job on behalf of them and generate the results. The proposed system is demonstrated by C# (Server side) and Android application development tool (Client side).

The proposed article helps to understand the proposed system architecture and operation of mobile agents in the system.

***Index Terms*—e-business; mobile agents; mobile devices; multi-agent systems; wireless, android**

I. INTRODUCTION

With the enlargement of E-business application it is ease to exchange things online; it helps with buying/selling process from any ware or any time. Online process of buying and selling is restricted due to availability internet access due to the lack of connection. Feature of mobility of mobile devices cause a problem for internet access, due to wireless connections can’t be available for all the time and at all the places. Also mobile devices are suffering from less processing power and less storage capacity which restricts the use of mobile devices for e-commerce applications.

This article proposes a feasible architecture that combines agent mobility and intelligence for consumer-oriented e-business applications. It allows a user to create a mobile, intelligent agent via a mobile device, and then launch the agent to the Internet to perform business on the user’s behalf. The aspect of mobility enables our architecture to support the agent’s migration and the user’s mobility (the ability to conduct e-business via mobile devices anyplace and anytime). The mobile agent will migrate from market to market, communicating with different trading agents to find the most appropriate one. Once an appropriate agent is found, it will inform the user of the results. This architecture complements the current Web-based, Internet systems by adding the wireless channel of mobile agents. Our current work focuses on lightweight mobile agents who act on behalf of consumers and participate in consumer-to-consumer (C2C) e-business applications, business-to-consumer (B2C) or business-to-business (B2B) applications.

II. ADVANTAGES OVER THE CURRENT E-BUSINESS SYSTEM

Many people nowadays use mobile devices such as tablets or mobile phones to access information through the Internet. In addition, they desire to have the ability to participate in e-business anywhere and anytime via their mobile devices. Current e-business applications, such as business-to-consumer (B2C), consumer-to-consumer (C2C) or Internet-based shopping, are typically developed over the Web for human-computer interaction. These applications require that users must login the intended Web sites from their personal computers or public terminals. Also, users often need to visit lots of sites and are always involved in a time-consuming process. To address these challenges, several wired agents-based e-business systems have been proposed. [1], for example, is an electronic marketplace where buying and selling agents can carry out business on behalf of their owners. Nevertheless, these systems do not satisfy the users' mobile demand due to their lack of wireless channels.

III. Purpose System Architecture

The term shopping comes with multiple ways to shop for goods, shopping divides into two things; i.e. fresh / new products and second hand products. For buying a new product customer can chose e-shopping or buy that product from nearest shop in case of urgency. Bidding for resale product or buying a fresh product online as per the requirement is normal but if we give a facility to buy a product from a nearest shop by checking the availability of product online which will reduce the time and efforts for searching the product in multiple shops.

All the above mentioned shopping preferences are making available to the customer by our system.

Figure 1 describes the system architecture and its working procedure. As shown in figure 1, users can interact with the system with mobile devices or desktop computers. Mobile devices are supported by the personal agents and connected to the mediator server via wireless connections.

A personal agent is a static agent running on a mobile device and offers a graphical user interface (GUI) for its user to communicate with the system. The mediator server is connected to the Internet where other mediator servers or other FIPA[1]-compliant systems exist. In the mediator server, an Agent platform answers any requests from the personal agent[2] and is linked to the behavior of a proxy agent[3] in charge of handling the requests. The proxy-agent interfaces with the Agent platform and constructs a bridge between the Web service server and the multi-agent system. Each instance of the behavior connects not only to the AMS agent (Agent Management Service as defined in FIPA, i.e., the white-page agent mentioned above), asking for the creation of a buying or selling mobile agent in the multi-agent system as well as providing a response, but also connects to the agent DF[4] (Directory Facilitator as defined in FIPA, i.e., the yellow-page agent), retrieving the list of agent advertising services with the DF. In this architecture, the multithreaded-Agent platform server is mirrored by a multi-behavior proxy agent to allow for handling multiple requests in parallel.

As illustrated in Figure 1, the procedures from (1) to (7) depict how a buying or selling mobile agent is created by a user according to preferences:

1. At the first step, the user selects a mode of shopping and then user configures the preferences via the personal agent (residing on the mobile device).
2. The personal agent then sends an XML-based request to the mediator server.
3. An instance of the Agent platform accepts the request and communicates with the proxy agent through the Agent Platform.
4. The proxy agent cooperates with the AMS agent who lives in the main container of the middleware platform to create a buying or selling mobile agent.
5. If the buying or selling agent is created successfully in the container, it might be mobilized to other systems to undertake the user's task.
6. And (6) the personal agent receives a response from the proxy agent via the Agent platform and informs the user of the relevant mobile agent being created.

The above is an asynchronous process after which the user can disconnect from the network at will. Even if the user decides to disconnect from the network, the user will still receive an SMS-based notification from the mediator server via an interface with the wireless carrier, or an e-mail-based notification from the mediator server via an interface with a mail server, as long as the user reconnects to the network.

The mediator server provides the required support for the creation of mobile agents, messaging among agents, agent migration facility, collaboration, protection, destruction, and control of mobile agents. Obviously, any multi-agent system can be used here as long as it provides the required support.

IV. Mobile Agents Behavior

As illustrated in Figure 2, a mobile (buying or selling) agent starts with its registration in the system and ends with a timeout of its lifetime. There are three time events that indicate the behaviors of a mobile agent: (1) the agent starts its negotiation process at a regular interval (e.g., every minute); (2) the agent starts its migration when activity time per server is reached; and (3) the agent ends its life cycle when its lifetime is exhausted. An argument may arise; how can one be sure that the mobile agent will be terminated according to the parameter and lifetime, as users prefer? This parameter may be changed by a third party (including the mediator server). The assumption we made is that the mobile agent can be protected from the attacks (e.g., from the host or other agents) once a future security mechanism is imposed on our architecture.

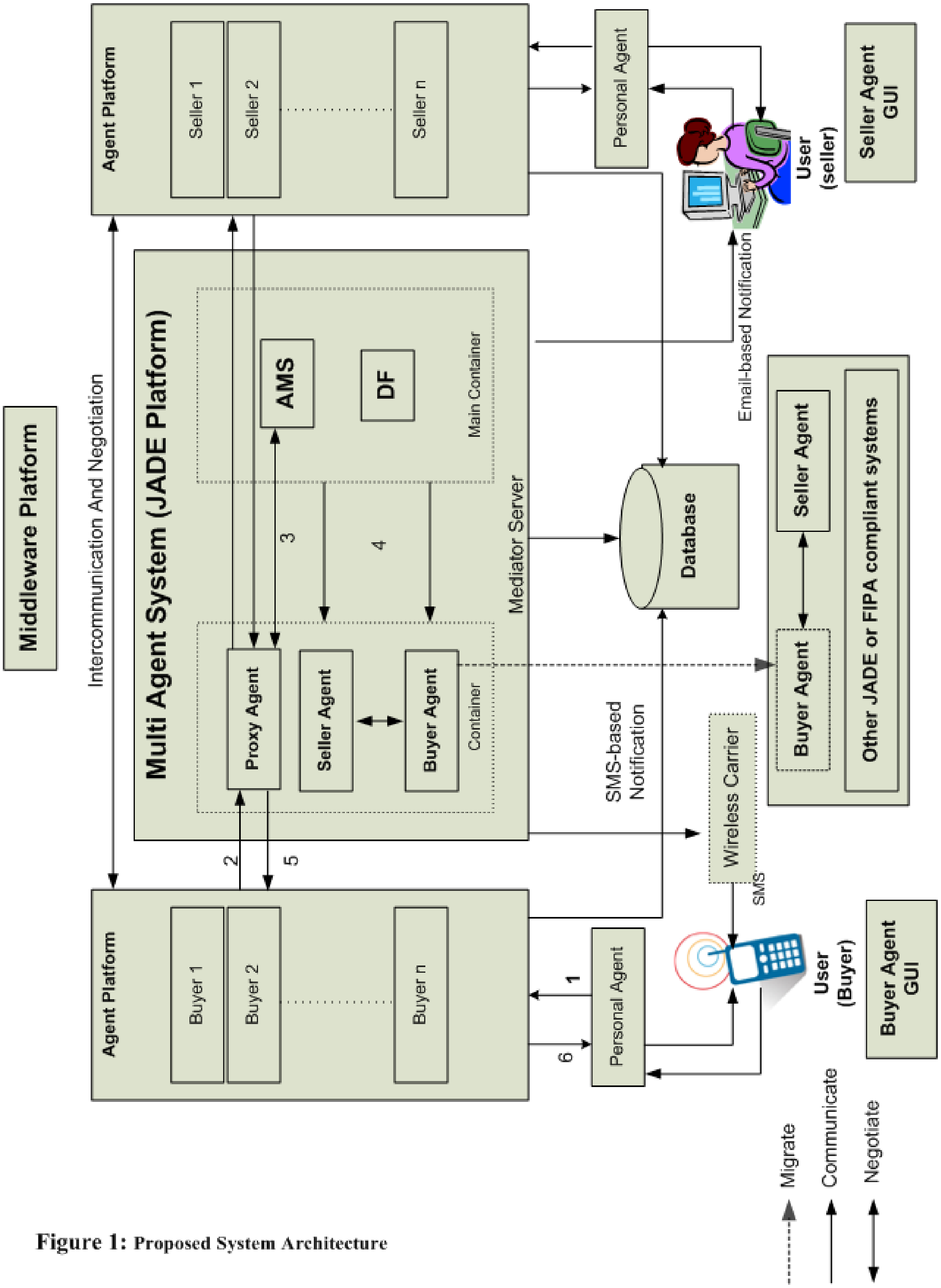

Figure 1: Proposed System Architecture

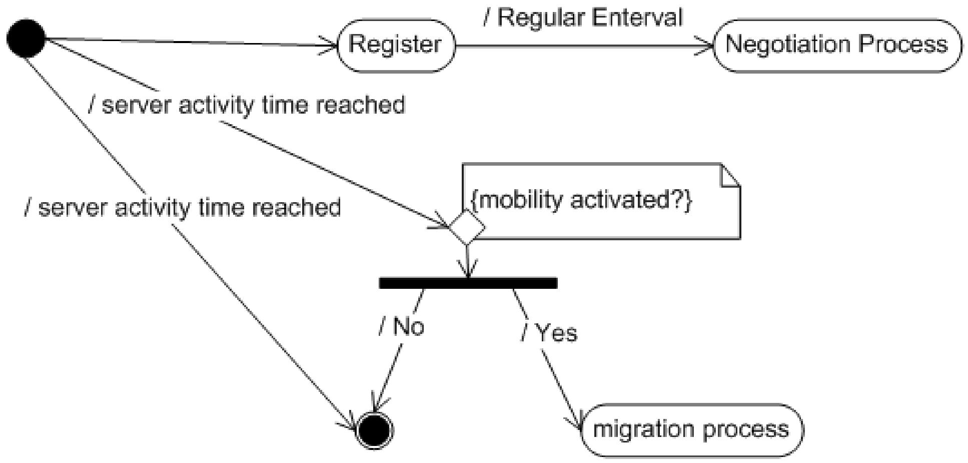

Figure 2: Activity diagram of mobile agent

V. Scope of Project

This system proposes a feasible architecture that combines agent mobility and intelligence for consumer-oriented e-business applications. It allows a user to create a mobile, intelligent agent via a mobile device, and then launch the agent to the Internet to perform business on the user's behalf. The aspect of mobility enables our architecture to support the agent's migration and the user's mobility (the ability to conduct e-business via mobile devices anyplace and anytime). The mobile agent will migrate from market to market, communicating with different trading agents to find the most appropriate one. Once an appropriate agent is found, it will inform the user of the results. This architecture complements the current Web-based, Internet systems by adding the wireless channel of mobile agents. Our current work focuses on lightweight mobile agents which act on behalf of consumers and participate in consumer-to-consumer (C2C) e-business applications.

Since personal software agents essentially need to communicate with other agents (to accomplish their designated tasks), they have to comply with a set of standards concerning the agent communication language and the protocols to be used. Although there is currently no universally accepted set of standards for developing multi-agent systems, the Foundation for Intelligent Physical Agents (FIPA), which aims at providing one language commonly understood by most agent-based systems [2], is obtaining a growing acceptance.

Multi-agent technology which involves intelligent agents should help facilitate e-commerce processes. Furthermore, it can be expected to have an important economic impact, by bringing efficiency to businesses (and thus improving their profitability) as well as benefiting individual users [3].

The Middleware based Agent systems provide a platform for building such system where heterogeneity is no longer a problem and communication is possible among components of the system independent of the platform, and can be distributed with success over the Internet.

With the use of agent oriented middleware, we can reach our main goal which is making electronic commerce (E-commerce) much easier and more successful.

Software agents are of good use to reduce work and information overload and provide high quality services to humans. With software agents, routine electronic transactions can be automated to a certain extent and hence human involvement is reduced.

Software agents should be conceptualized and implemented as independent agents, where each agent is responsible for a particular sub-function of the system [4].

For communicate with each other for those agents, it is desirable for them to have a common communication language and to follow common protocols.

In order for these agents to be able to cooperate, we need a middleware service which will provide a common background of interaction among agents.

If agents are to communicate in a way that, they must share the same vocabulary, language and protocols. They mainly communicate by exchanging messages following FIPA specifications [2].

The process of buying something online can be divided into two major phases: first one has to find the best place (host) to buy certain goods or services, and then one has to pay for the goods or services that are bought. This report investigates how both phases can be conducted securely and efficiently by mobile agents using certain cryptographic techniques.

VI. Motivation

It should be noted that mobile devices suffer not only from limited battery time, memory, and computing power, but also from small screen, cumbersome input, and limited network bandwidth and network connection [5]. The proposed architecture, by making use of mobile agent technology, offers a solution to those problems. That is, after creating and initializing a mobile agent to act on the user's behalf, a user can disconnect the mobile device from the server. The user only needs to reconnect later on to recall the agent for results, hence minimizing the use of resources. In addition, mobile agent technology also addresses such challenges as increased need for personalization, high latency, demand for large transfers, and disconnected operation [6].

The mobile agent paradigm is applicable in many application fields from which some areas are specially recognized as e-commerce, electronic auctions and stock market. These areas need high transaction of money. Now the user will put his money in mobile-agent only when he is confident that his money is secure and the agents can be trusted which are dealing with his money or transmitting some secure information. This is one reason which instigates much research effort in security of mobile-agents which has its own benefits when used in these areas.

As the name suggests that these agents are mobile in nature, which make them to travel autonomously in the network, due to this nature the agents become more vulnerable to various attacks. These attacks expose the limitation of mobile agents in the security area.

VII. Usecase Model of System

Figure 3 shows the use case model of our system. This use case depicts the different roles played by actors in the System. Actors include 1) human users (buyers, sellers), 2) software agents (Proxy Agent, Buyer Agents, Seller Agents, Personal agent)

The Buyer or customer is a person whose main buying tasks will be handled by the buyer agent. The seller or supplier is a person who will delegate the main selling tasks to the seller agent.

Buyers and sellers need to be registered to the system database in order to be authenticated and be allowed to proceed with trading process.

After successful authentication, the user can create buyer agents, if he is a buyer wishing to buy some products. He can create as many buyer agents as he wishes depending on the categories of products to buy.

The buyer agents will be given a task of buying specific products and start searching for seller agents who sell the needed products.

If seller agents are found, the buyer agent makes an offer to them to buy the product. This request is a call for proposal containing 1) the product name, 2) the product category and 3) the price.

This offer can be rejected by the seller agent if the offered price is low or if the product is not available for sale.

When the offer is accepted, the buyer agent is involved in price negotiation with the seller agents.

This negotiation, in this system, follows the logic of Dutch auction[7] where price negotiations continues until the buyer agent completes its negotiations, and ends when the product is sold or the reservation price of the seller agent is reached.

When an agreement is reached after negotiating, the buyer agent places an order, and if the order processing is successful, the product is sold to the buyer agent.

The Seller needs to have his products registered to the database in order to sell them. The personal agent is the one responsible of informing the status of the inventory to the seller agent.

The seller agent starts now selling the products it is asked to sell by the seller user. This seller agent has the capacity of handling all offer requests from the different buyer agents. It rejects the offer if the product is not available. If the product is available for sale, the seller agent now proposes to the buyer with the price.

The negotiation starts. As already mentioned above, this negotiation is following the logic of Dutch auction [4].

This negotiation process is not only done as a one to one scenario. The buyer agent is able to participate in multiple auctions for a given good, placing bids appropriately to secure the cheapest price [7]. The seller agent is also able to be in intense negotiations in order to sell the product to the buyer who offers a good price.

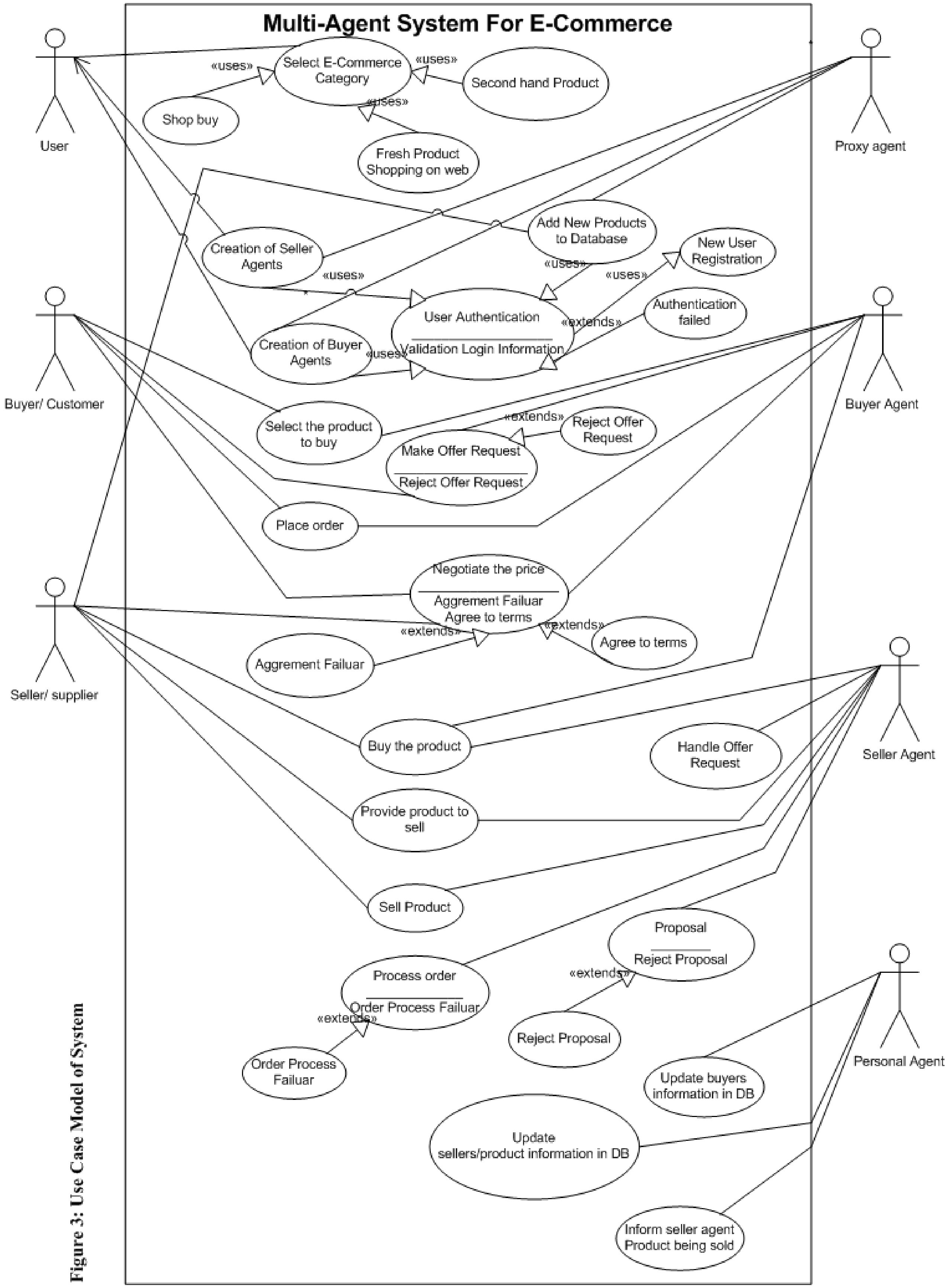

Figure 3: Use Case Model of System

VIII. System Implementation

For the buyer's emulator, the user activated the mobility of the buying agent, but it was not the case for the seller's emulator. We observed the following results:

- Mobile users can connect to mediator servers via HTTP and initiate mobile buying or selling agents in the mediator server.
- Mobile users do not need to instruct their mobile agents of what to do after configuring their preferences.
- Mobile users can add new items anytime and relevant mobile agents will be created to handle the trading of these new items respectively.
- Mobile users can kill their mobile agents to cancel their tasks by sending instructions to their personal agents.
- Mobile agents are active in their servers within the specified server activity time and then migrate to other servers.
- Buying agents can reach agreements with selling agents when the required item and price are matched. Mobile users then receive text messages from their agents, displayed on the screen of the simulators.
- Mobile agents end their life cycles when finishing their tasks.

As confirmed by the experiments, mobile users connect to their servers only when they need to add new items or to cancel their tasks. This obviously results in such benefits as reduced bandwidth utilization, increased battery life for mobile devices, and no complicated computation conducted on mobile devices. Also, mobile agents can move to various servers to negotiate autonomously, and mediator servers can accept mobile agents from outside their systems. This feature enables users to participate in multiple markets on the Internet.

IX. FUTURE WORK

Currently, we present a conceptual framework that needs to be refined. Using this work as a starting point, we have outlined a number of future research directions:

(1) Negotiation protocols do not have to be hard-coded into the agents. Instead, mobile agents can adapt to any intelligent negotiation strategies when they arrive at a new remote location. Thus, our architecture paves the way for future research in which more general architectures can be explored to allow mobile agents to participate in a variety of negotiation protocols, such as factor negotiation (price, quality, delivery time, etc.), electronic contracting, and so on. Currently, the negotiation strategy module consists of only a purchase determined by price (agents seek a preferable price by a fixed amount). FIPA defines auction protocols (e.g., Dutch and English auctions) as well as simple strategies such as fixed price, fixed price with a discount, and so on. We will add them into the negotiation protocols in our future research.

(2) The items are described only by their names. Obviously, other attributes, such as color, age, terms of warranty and delivery should also be considered. We believe that ontologies can help to solve this problem. It should be noted that the small screen of mobile devices will bring inconvenience to users when they specify many attributes of an item. A possible solution is to make use of the persistent memory of mobile devices to store the users' preferences.

(3) Mobile agent technology currently has some limitations, such as identity management, fault tolerance, protection of agents, and resource security. These limitations have brought up some concerns about the practical utilization of mobile agents. For example, in the area of security, e-business applications are often involved with money and thus users may hesitate to use mobile agents, unless mobile agents are secure enough to be trusted.

In the situation presented in this article, the mobile agents representing different buyers or sellers migrate over the Internet and then execute themselves on remote computers. These mobile agents are thus exposed to open environments and may become vulnerable. Since the mobile agents execute on unknown computers and interact with unknown agents, a reliable security infrastructure is vitally needed for the design of the system. The mobile agents must be able to deal with situations where they have been shipped off to the wrong address or to a hostile environment [8]. Listed below are some possible security concerns:

- Malicious mobile agents can try to access services and resources without a malicious agent may assume the identity of another agent in order to gain access to platform resources and services, or to cause mischief or even serious damage to the platform.
- Mobile agents may suffer eavesdropping attack from other mobile agents. A malicious agent can sniff the conversations between other agents or monitor the behavior of a mobile agent in order to extract sensitive information from it.
- Mobile agents may suffer an alteration attack from malicious hosts. To execute the agent and update its state, the host must definitely be capable of reading and writing the agent. A malicious host may

steal private information from the agent or modify the agent to compute the wrong result or to misbehave when it jumps to another site.

Current research efforts in the area of mobile agent security adopt two different perspectives [9]: First, from the platform perspective, we need to protect the host from malicious mobile agents (such as viruses and Trojan horses) that are visiting it and consuming its resources. Second, from the mobile agent perspective, we need to protect the agent from malicious hosts. There are many mechanisms to protect a host against malicious agents. Digital signatures and trust management approaches may help identify the agent and evaluate how much it should be trusted. The malicious host problem, in which a malicious host attacks a visiting mobile agent, is the most difficult problem. We found in the literature some works on powerful techniques such as Sandboxing and Proof-Carrying Code (PCC). Sandboxing [10] is a software technique used to protect a mobile agent platform from malicious mobile agents. PCC [11] introduces the technique in which the code producer is required to provide a formal proof that the code complies with the security policy of the code consumer. Therefore, we envisage that the security of mobile agents is an important issue that will encourage techniques and mechanisms for e-business in the future.

X. CONCLUSION

We propose in this article an e-business architecture that allows traders to do business in remote locations by means of mobile intelligent agents. Our architecture, which adheres to standardization efforts in the multi-agent field such as FIPA paves a possible way towards a near future when mobile buying (and selling) agents can smoothly travel among different agent-based marketplaces to carry out tasks on their users' behaviors. Our purpose of presenting this idea is to improve our understanding of the value of mobility and to encourage the conceptual construction of a global community. We do not claim that buyers and sellers around the world would have to buy into this to make it work, and that worldwide C2B, B2C, B2B e-commerce would be revolutionized thereby. In practice, however, we hope that our work would be useful on a smaller scale and lead to new investigations that may result in new solutions to the problems we addressed. Our proposed architecture, aimed at providing new capabilities for advanced e-business solutions, employs an approach that integrates intelligent and mobile agents. Intelligent agents can provide automation support for decision-making tasks, while mobile agents can extend that support by allowing users to participate in several marketplaces in a networked e-business. We believe that intelligent and mobile agent technology is also a promising solution to the problems of low speed, high latency, and limited computing ability that the current wireless network is facing.

APPENDIX

1. FIPA-
2. PERSONAL AGENT-
3. PROXY AGENT-
4. DF-

REFERENCES

[1] Chavez, A., & Maes, P. (1996). Kasbah: An agent marketplace for buying and selling goods. In Proceedings of the 1st International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, United Kingdom.

[2] FIPA. (2006). Retrieved May 12, 2013, from http://www.fipa.org

[3] Amalia Pîrvănescu, Costin Bădică, Maria Ganzha, Marcin Paprzycki (2005). Conceptual Architecture and Sample Implementation of a Multi-Agent E-Commerce System (pdf) In: Proceedings of the 15th International Conference on Control Systems and Computer Science CSCS'15, Editura "Politehnica Press", Bucharest, 2005, pp.620-616

[4] Costin Bădică,Maria Ganzha, Maciej Gawinecki, Pawel Kobzdej, Marcin Paprzycki (2006) Utilizing Dutch Auction in an Agent-based Model E-commerce System. In: Proceedings of the 14 th International Enformatika Conference, World Enformatika Society, 2006, 7-12

[5] Wang, A.I., Sørensen, C.F., & Indal, E. (2003). A mobile agent architecture for heterogeneous devices. Wireless and Optical Communications.

[6] Kotz, D., & Gray, R. (1999). Mobile code: The future of the Internet. In Proceedings of Autonomous Agents'99: Workshop on Mobile Agents in the Context of Competition and Cooperation.

[7] Finin, T., Y. Labrou, and J. Mayfield, "KQML as an Agent Communication Language", In Book: Software Agents, J. Bradshaw, MIT Press, (1997).

[8] Neuenhofen, K.A., & Thompson, M. (1998). Contemplations on a secure marketplace for mobile Java agents. In K.P. Sycara & M. Wooldridge (Eds.), Proceedings of Autonomous Agents 98, Minneapolis, Minnesota. New York: ACM Press.

[9] Kotz, D. (2002). Future directions for mobile agent research. IEEE Computer Science.

[10] Wahbe, R., Lucco, S., Anderson, T.E., & Graham, S.L. (1993). Efficient software-based fault isolation. In Proceedings of the 14th ACM Symposium on Operating Systems Principles (pp. 203-216).

[11] Lee, P., & Necula, G. (1997). Research on proof-carrying code on mobile-code security. In Proceedings of the Workshop on Foundations of Mobile Code Security.

Kalpana N. Meher Perusing M.E. Computer Engineering from MGM's College of Engineering and Technology, Navimumbai, India. Working as Lecturer at MGM's College of Engineering and Technology, Navimumbai, India. Having 4 years of Teaching Experience.	
Prof. Sanjay Jadhav Working as Assistant Professor in Computer Engineering at Saraswati College of Engineering Navi mumbai, India. Having 8 years of Teaching and Industry Experience. Published more than 25 paper in various National, International conferences and Journals. His basic area of interest is Software Engineering, E-Commerce, E-Commerce Security, Information and Network security, Operating system etc.	
Prof. P S Lokhande Working as Head Dept of IT at MGM's College of Engineering and Technology, Navimumbai, India. Having 14 years of Teaching and Industry Experience. Published more than 20 paper in various National, International conferences and Journals. His basic area of interest is Web Engineering, E-Commerce, E-Commerce Security, Digital Evidence Collection etc.	