EXPLORING MOBILE-AGENT-BASED ARCHITECTURES FOR M-COMMERCE APPLICATIONS

Gilda Pour
San Jose State University
San Jose, CA, U.S.A.

ABSTRACT

Mobile commerce (m-commerce) has emerged as the promising approach for driving the next wave of e-commerce. A major issue facing m-commerce is the lack of flexible, adaptable, extensible, and robust architectures for mobile and pervasive application systems for m-commerce. To address this issue, our research[1] has explored how to employ mobile agents in developing m-commerce applications. We have developed a new mobile-agent-based architecture to help advance creating new m-commerce applications. This paper presents the new mobile-agent-based architecture and a demonstration prototype of inventory management and tracking systems with the new architecture for mobile environments. The prototype is built using J2ME, J2ME-based technologies such as CLDC and MIDP, and Java-based mobile agent technologies such as JADE-LEAP.

KEYWORDS

Mobile commerce, mobile-agent-based architecture, agent components, J2ME, JADE-LEAP

1. INTRODUCTION

Mobile commerce (m-commerce) has emerged as the promising approach for driving the next wave of e-commerce. Jupiter Media Matrix has projected a major increase in the mobile commerce revenues from $0.4 billion global revenue in 2000 to $31+ billions worldwide in 2005. The projection for 2005 includes $3.3 billions in U.S.A., $3.5 billions in North America, $7.8 billions in Western Europe, $9.4 billions in Asia, $5.5 billions in Japan, $0.5 billion in Latin America, and $1.0 billion in other parts of the world.

A critical issue facing mobile commerce is the lack of flexible, adaptable, extensible, and robust architectures for pervasive and mobile commerce application systems. To address this issue, our research[2] has explored how to develop m-commerce application systems using mobile agent components. We have developed a new mobile-agent-based architecture to help advance creating new m-commerce applications.

This paper is organized as follows. Section 2 provides background information. Section 3 presents our mobile-agent-based architecture for m-commerce applications. It also provides an overview of mobile agent technologies for pervasive system development. Section 4 describes the development of a prototype of inventory management and tracking systems with new mobile-agent-based architecture. Section 5 presents final remarks and the future enhancement.

2. BACKGROUND

This research project has extended an earlier research effort led by the author [Pour and Guo 2003]. In the previous project, we used Web services and components as building blocks of the architecture. In this research project, we have employed mobile agent components to design new mobile-agent-based architecture

[1] The research has received sponsorship from IBM and Hewlett-Packard.
[2] Sponsored by IBM and Hewlett-Packard

for m-commerce applications. We have also developed a demonstration prototype of inventory management and tracking systems with the new architecture for mobile environments.

We view software agent components as loosely coupled, active, messaged-based distributed components. They offer greater flexibility and adaptability than traditional components [Pour (1) 2004, Pour (2) 2004, Pour (3) 2004, Pour and Wei 2004, Pour et al (1) 2004, Pour et al (2) 2004, Pour 2003, Griss and Pour 2001]. Agent components with the following characteristics are more suitable than traditional components for developing architectures for m-commerce applications:

- Mobile – The agent is able to move from one executing context to another, either by moving its code and starting a fresh or by serializing its code and state, continuing execution in a new context and retaining its state to continue its work.
- Autonomous – The agent has an agenda of goals; it proactively watches for relevant situations (events), and performs actions based on an assessment of the situation.
- Highly customizable – The agent is driven by the user goals and tasks.
- Collaborative – The agents seek other agents to perform sub-goals or provide information [Griss and Pour 2001, Wooldridge et al 1999, Jennings 1998].

We refer to agent components as agents in this paper. Agents can collaborate with one another in a flexible, dynamic way, using rich message-based communication. Agents are used to vigilantly monitor business process and system status, and flexibly adapt system response and reconfiguration to circumstances.

Agent technology has matured significantly in recent years. FIPA standards (www.fipa.org) define an agent platform architecture that supports agent communication, agent management, and agent message transport. FIPA specifies languages, message sequencing, ontology and protocols, agent management and standard agents such as the Facilitator.

FIPA-compliant agent platforms are available as open source in Java (e.g. JADE [Bellifemine 1999]). JADE agents have been integrated into J2EE application servers to provide a more robust and scalable agent platforms (e.g. BlueJADE [Cowan and Griss 2001]). There have been numerous applications of agent components in research as well as in advanced development and products [Pour (1) 2004, Pour (2) 2004, Pour (3) 2004, Pour et al (1) 2004, Pour et al (2) 2004, Pour and Wei 2004, Pour 2003, Griss and Pour 2001].

3. THE ARCHITECTURE DESIGN AND DEVELOPMENT

To address the deficiencies of the existing systems, we have developed a mobile-agent-based architecture for mobile commerce applications. The primary objective of this research project is to explore and extend the benefits of agent components (e.g. mobility, autonomy, collaboration and high customizability) to m-commerce applications.

It is crucial to address flexibility, adaptability, extensibility, maintainability, robustness and manageability in the design of architecture. To do so, we have adopted agent-oriented software engineering approach to design an agent-based architecture for m-commerce applications in mobile and pervasive computing environments.

The use of mobile agent components in the m-commerce application systems helps build more resilient systems as mobile agents can migrate away from failed parts of the system as well as more flexible and adaptable systems to support highly dynamic environments [Pour et al (1) 2004].

3.1 High-Level Design

To address the need of mobile users of m-commerce applications, the system has to logically follow the users from one environment to another, and be available to them at any time and any place. Such level of mobility requires refocusing the software system from large, monolithic applications to collections of mobile agents to allow processing users' requests by performing a series of tasks. This research project makes use of mobile agents to support mobility and wireless capability m-commerce applications.

Figure 1 shows high-level design of our mobile-agent-based architecture. The mobile-agent-based architecture is designed with three main layers: (1) front layer, (2) middle layer, and (3) back layer. The front layer is mainly for presentation, the middle layer for application logic, and the back layer for data management.

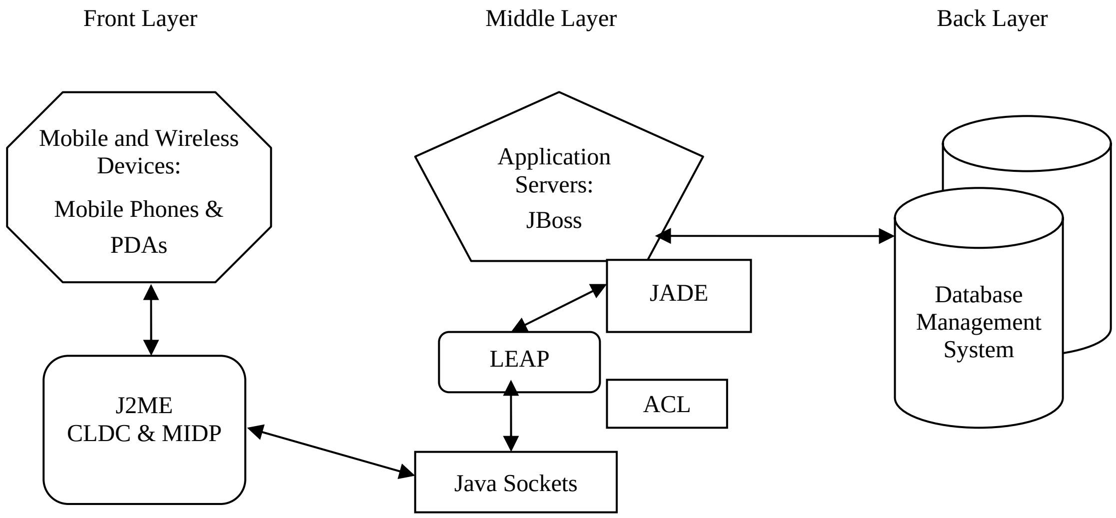

Figure 1. High-Level Design of the Mobile-Agent-Based Architecture

3.2 Types of Agents

The new architecture contains the following main classes of agents:

- User interface agents
- Process agents
- Wrapper agents

User interface agents receive requests from users (humans and other systems), package the requests and send them to appropriate process agents. Receiving and the shipping agents are examples of user interface agents. User interface agents can be passive or proactive. Proactive form is more suitable for intelligent systems. The passive form is widely used as it produces less ambiguous direction for processing requests.

Process agents translate the users' requests that they receive through user interface agents into a series of tasks; determine the best sources to obtain the information required for processing the requests, and make decisions among alternatives. These agents act as surrogates for users in managing and executing the process. Tracking agents is examples of process agents.

Wrapper agents represent legacy systems in the mobile-agent-based architecture. Each legacy system is represented by a wrapper agent component (a.k.a. wrapper agent). A wrapper agent represents a legacy system's interfaces, announces the legacy system's capabilities in agent compatible ways, acts as the proxy of the legacy system, handles all service requests intended for the system, and manages the system interactions.

3.3 Underlying Agent Technologies

In the front layer, Java 2 Platform, Micro Edition (J2ME) technologies such as the Connected Limited Device Configuration (CLDC) (http://java.sun.com/products/cldc/) and the Mobile Information Device Profile (MIDP) (http://java.sun.com/products/midp/) are coupled. J2ME (http://java.sun.com/j2me/) provides an application environment that specifically addresses the needs of embedded and mobile devices such as cell phones and PDAs.

CLDC defines the base set of application programming interfaces and a virtual machine for resource-constrained devices like mobile phones and mainstream personal digital assistants. CLDC coupled with MIDP provides a solid Java platform for developing applications for mobile and wireless devices with limited memory, processing power, and graphical capabilities (e.g. mobile phones and PDAs).

JADE is a Java-based open source FIPA-compliant agent platform. JADE supports implementation and desktop deployment using an extensive developer's toolkit and compatibility with standard and 'heavier' Java environments [Bellifemine 1999].

LEAP (http://leap.crm-paris.com/) is an agent platform for small mobile devices such as PDAs and cell phones. It is lightweight; extensible in size and functionality; operating system agnostic; supporting wired and wireless communications and FIPA-compliant. LEAP has reduced footprint and compatibility with mobile Java environments. LEAP is a library.

JADE-LEAP is the combination of Lightweight Extensible Agent Platform (LEAP) and Java Agent DEvelopment Framework (JADE). JADE-LEAP supports agents run on lightweight devices. It can be deployed on different types of devices, networks and JVMs. Figure 2 shows the JADE-LEAP run-time environment. JADE-LEAP offers a homogeneous set of APIs and ultimately allowing ubiquitous deployment of agent-based services and applications.

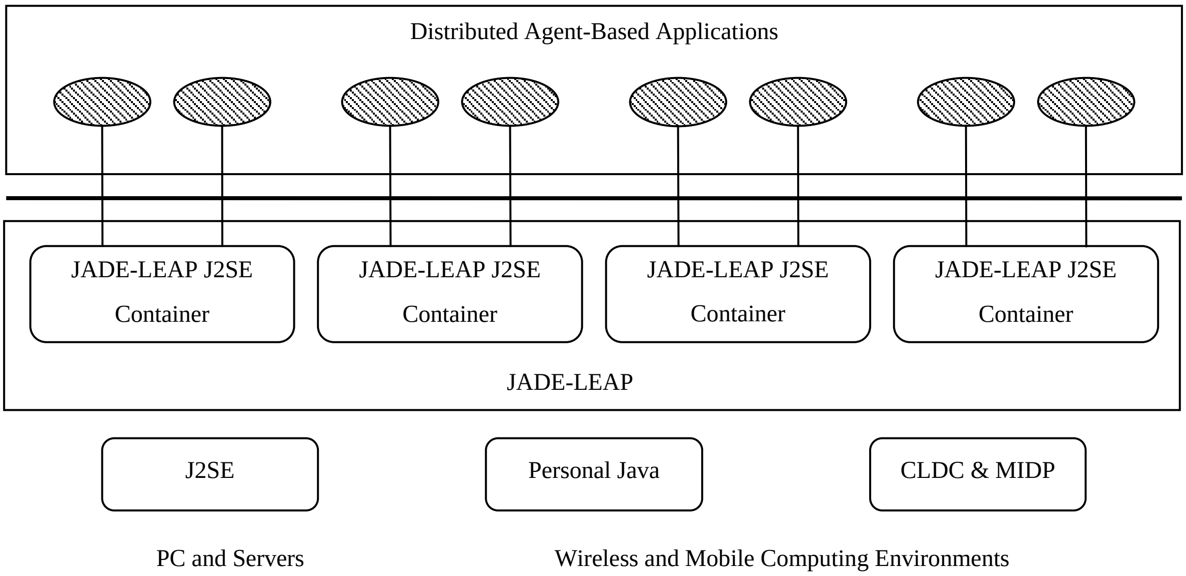

Figure 2. The JADE-LEAP Runtime Environment

As shown in Figure 2, JADE-LEAP allows agents run on lightweight devices in three different ways: (1) “Personal Java” to execute JADE-LEAP on handheld devices supporting Personal Java (e.g. PDA); (2) “MIDP” to execute JADE-LEAP on handheld devices supporting MIDP (e.g. cell phones); and (3) “J2SE” to execute JADE-LEAP on PC and servers [Pour at al (1) 2004].

“Split” execution model is used in the JADE-LEAP runtime environment on a handheld device. Figure 3 illustrates “split execution mode.” The “split” execution model is to split a container into two parts, permanently connected to one another: front layer and back layer. This would allow the use of a lightweight container at the front layer as it is required for handheld devices. Furthermore, a smaller number of bytes are transmitted over the wireless connection.

The middle layer consists of two main classes of components: Java sockets and ClientAgent. Java sockets are one end-point of a two-way communication link between two programs running on the network. Java socket classes represent the connection between a client program and a server program. ClientAgents are highly collaborative, mobile agents. A ClientAgent is a JADE-LEAP agent. Once a new user logs onto the system, a ClientAgent will be automatically created to act on the behalf of the user. The name of the ClientAgent is unique, and depends on the user ID. A ClientAgent holds the personal information of its user, and reacts to various incoming and outgoing messages and requests intended for the user. Once the user logs out, the life of ClientAgent comes to an end [Pour et al 2004]. Once the server receives a command, it finds the command type and parameter, generates a SQL command using the parameter, and communicates with a database. After the results are returned by the database, the server organizes the data and sends it back to the handheld device. Once the system is launched, the container automatically creates a back layer container to prepare for the other management activities. If more than one user logs onto the system simultaneously, a set of ClientAgents will be created, one for each user.

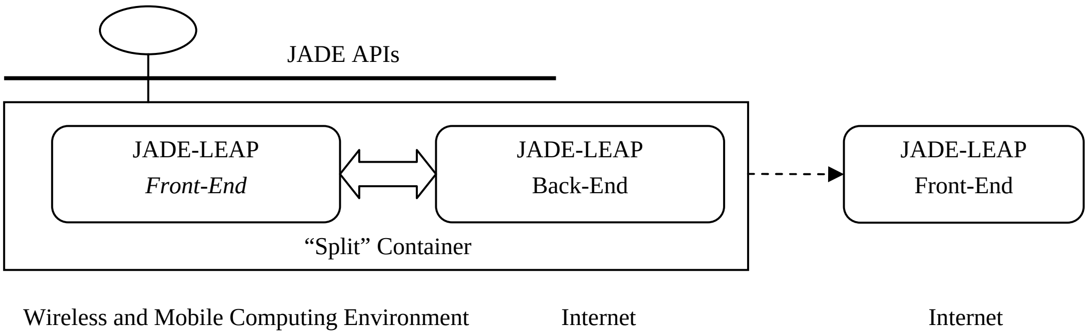

Figure 3. "Spilt" Execution Mode in JADE-LEAP

To address the security issue, we have taken the following measures: (1) An authenticated user has its own agent; (2) An agent acting on the behalf of its user has certain permissions; and (3) An agent has a signed certificate and the platform checks the agent's certificate to prevent unauthorized agents from taking actions that they are not authorized to take.

4. DEMONSTRATION PROTOTYPE DEVELOPMENT

We have developed a demonstration prototype of inventory management and tracking systems using the new mobile-agent-based architectures for m-commerce. We have used the ontology that we had developed for this application domain in our earlier research project. Agents use the FIPA/JADE AMS (white page) and Facilitator agent (yellow page) for registration and lookup. The capability of dynamically registering with the Facilitator agent allows new agents with new abilities to be introduced dynamically and incrementally. In this project, new TrackingIntegration and InventoryIntegration ontologies as well as TrackingOperation and InventoryOperation languages were developed and enhanced to support agent interactions for m-commerce systems. Figures 4 and 5 illustrate use case models for the tracking system and the inventory management system, respectively.

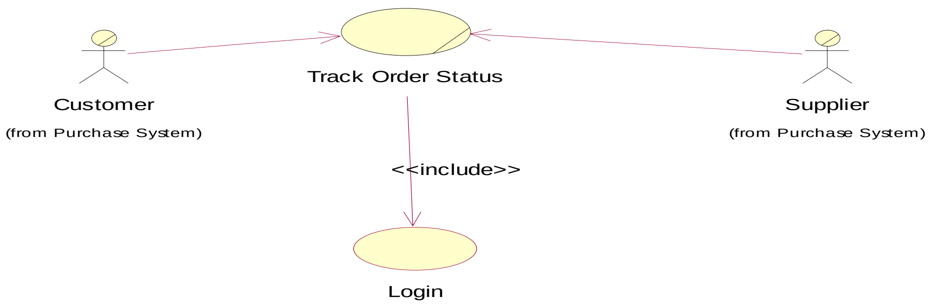

Figure 4. A Use Case Model for Tracking System

ISBN: 972-98947-8-7 © 2004 IADIS

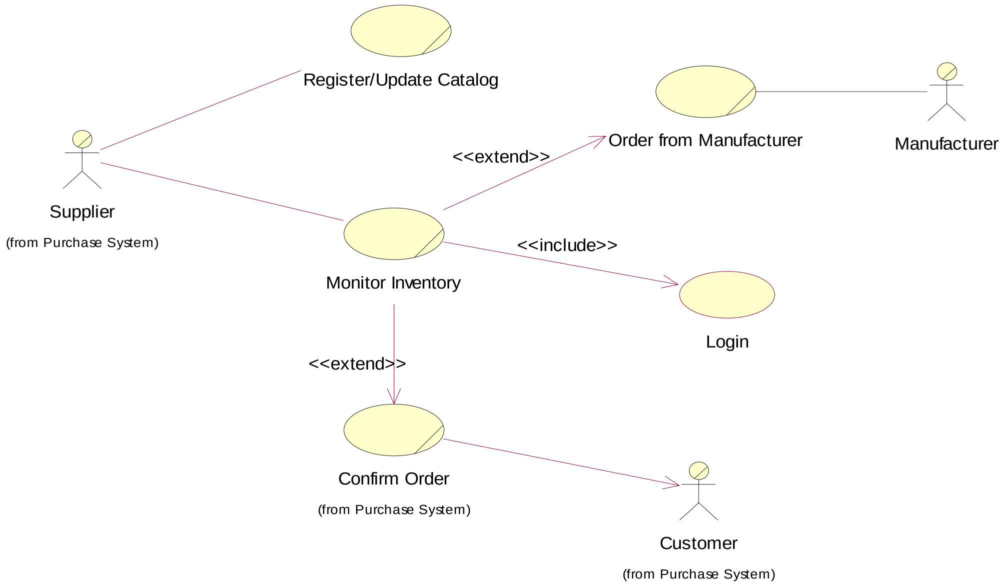

Figure 5. A Use Case Model for Inventory Management System

Figures 6 shows a sequence diagram for an inventory management system. Figures 7 and 8 illustrate a set of screenshots from the demonstration prototype.

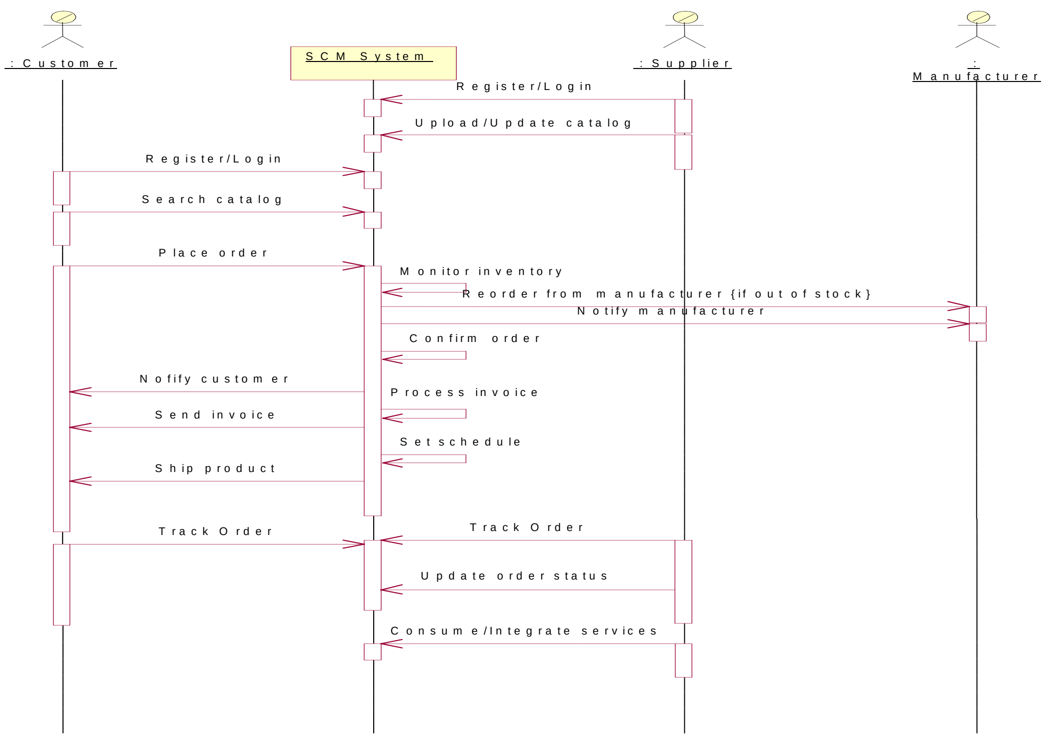

Figure 6. A Sequence Diagram for Inventory Management System

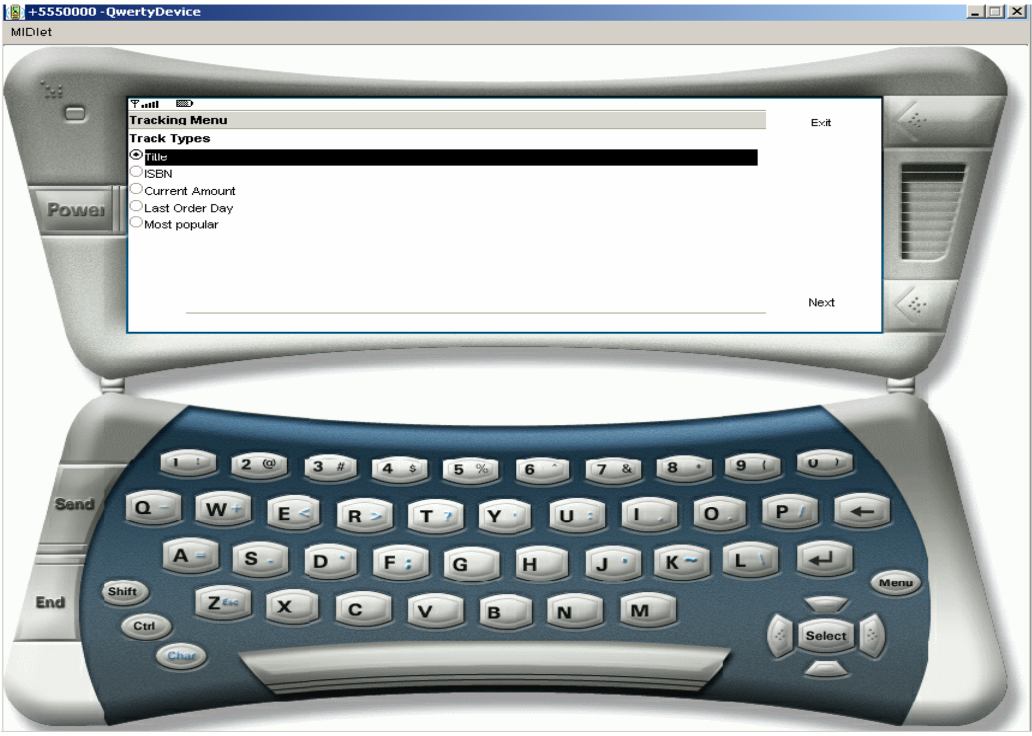

Figure 7. Screenshot of Order Tacking Menu on a Mobile Device

Figure 8: Screenshots of Order Tracking on a Mobile Phone

5. FINAL REMARKS

We have explored the development of m-commerce applications using mobile agent components. We have also developed a mobile-agent-based architecture for m-commerce applications, and built a demonstration prototype of inventory management and tracking systems with the new architecture for mobile environments. For the prototype development, we have used a variety of Java-based technologies and tools, such as JADE and J2ME, J2ME-based technologies including CLDC and MIDP, JADE-LEAP, JBoss and Jakarta Ant.

The new trend is to use mobile agent components for development of mobile and pervasive applications since mobile agent components support not only mobility but also higher level of flexibility, adaptability, flexibility and robustness than other alternatives (e.g. traditional software components [Pour 2000]).

The results of our applied research are promising. Hence, this project and its results can serve as the foundation for further research on development of mobile-agent-based architectures for m-commerce applications. The future directions for the research include exploring the possibilities for enhancing the current research in the following three ways:

(1) Scalability for the next-generation computing (i.e. pervasive computing);
(2) Fault tolerance (e.g. in the case of main container); and
(3) Security (e.g. which security information can be exchanged, how and when).

ACKNOWLEDGEMENT

The author gratefully acknowledges the sponsorship of IBM and Hewlett-Packard Company.

REFERENCES

Bellifemine, F. et al, 1999. JADE: A FIPA-Compliant agent framework. *Proceedings of Practical Applications of Intelligent Agents and Multi-Agents.* pp. 97-108.

Cowan, D., Griss, M. and Burg, B., 2001. BlueJADE – A Service for Managing Software Agents. *Hewlett-Packard Technical Report HPL-2001-296*.

Griss M. and G. Pour, 2001. Accelerating Development with Agent Components. Cover feature article, *IEEE Computer*.

Jennings, N.R. and Wooldridge, M.J., 1998. *Agent Technology*. Springer.

Pour, G., 2004. Web-Based Multi-Agent Architecture for Distributed Collaborative Software Engineering. *Applied Computing*, 2004.

Pour, G., 2004. Internet-Based Multi-Agent Architecture for Service Delivery," *Internet Computing Series*. pp. 173-179.

Pour, G., 2004. Multi-Agent System Architecture for Quality Control: Moving Toward Autonomic Systems for E-Commerce. *E-Commerce*.

Pour, G. and X. Wei, 2004. A Multi-Agent Framework to Support Web-Based Collaborative Environment for Drug Prescription Management. *Intelligent Agents, Web Technologies, and Internet Commerce Series*. pp. 294-305.

Pour, G., Yao, F. and Yu, C, 2004. A Mobile Agent-Based Architecture for Mobile Systems Supporting Distributed Software Project Management. *IEEE SoftCOM*.

Pour, G., Nguyen, C. and Mangoba, R., 2004. Web Service-Oriented Architecture for Pharmacy Management Systems in Mobile Environments, *ICS.*

Pour, G., 2003. Web-Based Multi-Agent Architecture for Software Development Formal Peer Inspection. *Intelligent Agents, Web Technologies, and Internet Commerce Series*. pp. 208-219.

Pour, G. and Guo, Y., 2003. Web Service-Oriented Architecture for Supply Chain Management, *Internet Computing Seri*es. pp. 702-708.

Pour, G., 2000. Component Technologies: Expanding the Possibilities for Component-Based Development of Web-Based Enterprise Applications. In Furht, B. (ed.) *Handbook of Internet Computing*. CRC Press, pp. 133-156.

Wooldridge, M.J. et al., 1999. Software Engineering with Agents: Pitfalls and Pratfalls. *IEEE Internet Computing*, Vol. 3, No. 3, pp. 20-27.