A Multi-Agent Negotiation System with Negotiation Models Changeable According to the Bargaining Environment

Sung Ho Ha* · Dong-Sup Kim**

Abstract

Negotiation is a process of reaching an agreement on the terms of a transaction, such as price, quantity, for two or more parties. Negotiation tries to maximize the benefits for all parties concerned. Instead of using human-based negotiation, the e-commerce environment provides such an environment as adopting automated negotiation. Thus, choosing agent technology is appropriate for an automatic electronic negotiation platform, since autonomous software agents strive for the best deal on behalf of the human participants. Negotiation agents need a clear-cut definition of negotiation models or strategies. In reality, most bargaining systems embody nearly one negotiation model. In this article, we present a mobile agent negotiation system with reusable negotiation strategies that allows agents to dynamically embody a user's favorite negotiation strategy which can be preinstalled as a component in the system. We develop a prototype system, which is fully implemented in compliance with FIPA specifications, and then, describe the benefits of using the system.

Keywords : Mobile Agent, Multi-Agent, Negotiation Strategy, Negotiation Agent

Received : 2008. 11. 10. Final Acceptance : 2009. 03. 13.

※ This work was supported by the Korea Research Foundation Grant by the Korean Government (MOEHRD) (KRF-2006-013-D00280). It was also supported by the second stage of BK21, Graduate school of Business Administration, Kyungpook National University.
* Corresponding author, Associate professor, School of Business Administration, Kyungpook National University, e-mail : hsh@mail.knu.ac.kr
** Senior researcher and engineer, Institute for Enterprise Content Management, IDS & Trust Co., Ltd., e-mail : dongsup.kim@gmail.com

1. Introduction

Business-to-business electronic commerce (B2B EC) is a new way of trading on the increasingly popular computer network, Internet, and it has rapidly become a major player in the business market. B2B EC helps buyers and sellers reduce business costs and it enables the customized delivery of goods and services. As B2B EC emerges as one of the most important EC areas, electronic negotiation is becoming an important research subject.

Negotiation is a process of reaching an agreement on the terms of a transaction, including price and quantity, for two or more parties (Dignum and Corts, 2001). It tries to reach a deal to maximize the benefits for all parties concerned. Human-based negotiation, however, could be an expensive and non-optimal method, even though beneficial to the participants, due to its potential in trying to yield the best results.

Traditional human-based negotiation has several shortcomings. More time may be needed to reach a consensus, if some of the human parties involved do not concede in order to maximize their own benefits or goals. It could be difficult for novice negotiators to mimic experts who have skillful negotiation tactics. Therefore, automated negotiation is particularly useful in an e-commerce environment.

The negotiation process is competitive and in general, is entirely dependent of negotiation counterparts. Thus, choosing agent technology is appropriate for an automatic electronic negotiation platform, since autonomous software agents strive for the best deal on behalf of the human participants. In addition, mobile agents become key skill in distributing negotiation information (Ye et al., 2001; Jennings and Bussmann, 2003).

In order to achieve the best results, a negotiation agent needs a clear-cut definition of what capabilities it can provide—negotiation models or strategies. Currently, there are several types of negotiation models for automated bargaining and many researchers have designed systems using the models. Nevertheless, most bargaining systems embody nearly one negotiation model (Bartolini et al., 2004). In this article, we present MANS/RNS, a Mobile Agent Negotiation System with Reusable Negotiation Strategy that allows agents to dynamically embody a user's favorite negotiation strategy which can be preinstalled as a component within the system.

The MANS/RNS has a mobile agent system framework that allows agents to travel from one server to another on the network and it allows agents to negotiate autonomously and automatically with counterpart agents regarding offers. It mainly focuses on using mobile agents for negotiations in an e-marketplace and integrating negotiation capabilities—in particular, components that implement negotiation strategies—into the mobile agents. That is, the MANS/RNS has a mechanism which enables negotiating agents to combine pluggable strategy components.

This article describes a set of similarities and differences between the MANS/RNS framework and a popular mobile agent system, Nomad which have been integrated with either eAuctionHouse or eMediator. The benefits of using the

MANS/RNS, especially the benefits of improving buyers' and sellers' satisfaction while preserving the system's performance, will be explained.

2. Negotiations in B2B Electronic commerce

Electronic commerce systems allow buyers to purchase goods through the Internet. However, it becomes much difficult to trace all available sellers through the Internet. Hence, the buyer may be interested in delegating their purchase tasksincluding negotiations to a software agent. With the increasing importance of electronic commerce across the Internet, the need for agents to support both buyers and sellers in purchasing and selling goods is growing rapidly (Oliver, 1996).

The most basic form of electronic negotiation is fixed-price sale where by the seller offers their goods or services through a catalog at 'take-it-or-leave-it' prices. At present, auctions are the most vivid type of negotiation on the Internet, as conducted by eBay. Combinatory auctions are another form of negotiation that involves making bids on a combination of goods or services. Electronic negotiations can take a more complex form called bargaining. This involves making proposals and counter-proposals until an agreement is reached. The Object Management Group sees bargaining as a bilateral negotiation (one-to-one bargaining) or a multi-lateral negotiation (many-to-many bargaining) depending on how many parties are involved. Negotiations can be further classified as either distributive or integrative. In a distributive negotiation, only one attribute (e.g., price) is negotiable. In an integrative negotiation, multiple attributes of an item are negotiable (Kang and Lee, 1998).

According to several reference models, B2B EC generally contains three phases : Information, negotiation, and settlement. In the first phase, prospective buyers identify and evaluate their needs as well as alternative sources to fulfill them, while sellers arrange to provide their goods and identify potential buyers. Subsequently, prospective buyers and sellers negotiate the terms of a deal, which is finalized by a contract. Eventually, the contract is executed and the objects of the transaction are exchanged according to the conditions previously stipulated. There have been several approaches dealing with phases one and three, but only a few systems exist for automatic negotiation concerning the second phase. This includes MarketMaker from MIT Media Labs. Until now commercial systems exist which equip mobile agents with automatic negotiation mechanisms.

There are some requirements for negotiation within electronic commerce (van Bragt and Poutré, 2003; Paprzycki and Abraham, 2003). One is to formulate the negotiation. Ontology must cover all important attributes of a product, such as price, quantity, or quality. Each essential variable must be specified to the agent for a valid comparison. The second requirement is a negotiation process. The negotiation process, which uses software agents, typically goes through four major steps : Delegation of a negotiating authority to agents; the initialization of a negotiation task and the proposal of an offer by one of the parties; the evaluation of the proposal by others; and suggestions of a new counter-proposal if other parties do not accept the offer. This process may be repeated several times until both parties reach an agreement.

The last requirement is to determine appropriate system architecture (Guttman et al., 1998). Such a system must have the capability to use the Internet as a backbone for transmitting data to different negotiating parties. In addition, the architecture must have the ability to support both synchronous and asynchronous negotiations. The widespread acceptance of Java provides the essential groundwork upon which an agent-based framework can be built. There have been several agent models which have embedded mobile agent system architecture into their designs : Aglet, Agent Tcl (Gray, 1996), ARA (Peine and Stolpmann, 1997), Concordia (Wong et al., 1997), Mole (Baumann et al., 1999), Odyssey, TACOMA (Johansen et al., 1997), Voyager, SHIPMAI (Tkito and Karmouch, 2000), and Nomad (Sandholm and Huai, 2000; Sandholm, 2002). Each framework promotes a different concept in agent programming. The use of abstractions varies, as well as the actual capabilities provided by each class library, as do the benefits they bring (Vogler et al., 1999; Chung and Honavar, 2000; Kowalczyk and Van Anh Bui, 2000; Lai and Lin, 2001; Rebstock, 2001; Schoop and Quix, 2001; Chiu et al., 2002; Lin and Kuo, 2002; Srivastava and Mohapattra, 2003).

3. MANS/RNS

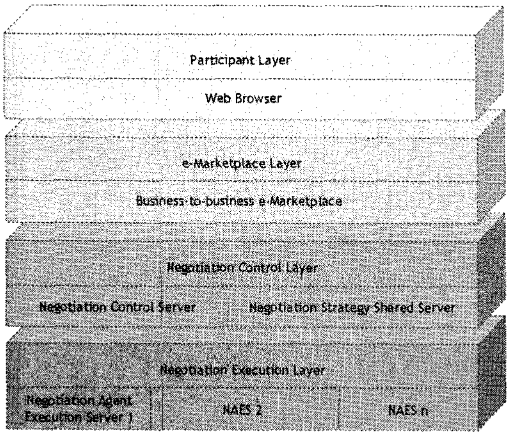
This study focuses on the negotiation phase in a B2B e-commerce transaction. We suppose that users, including buyers and sellers, have registered on an e-marketplace server, and that the information phase, such as developing specifications of goods and searching for merchants, has been successfully completed either automatically or semi-automatically.

In a normal situation, an e-marketplace incorporates many buying-agents and many selling-agents (N : N), regarding the purchase of a product or service. The buying-and selling-agents interact with each other simultaneously in order to negotiate multiple issues, including price or quantity.

3.1 Architecture of MANS/RNS

We have developed a mobile agent-based negotiation system with the reusing of adaptive negotiation strategies. <Figure 1> illustrates the high-level architecture of this system. The MANS/RNS is organized into a four-layer structure : participant layer, e-marketplace layer, negotiation control layer, and negotiation execution layer. Users in the participant layer visit an e-marketplace to trade goods or services. The e-marketplace layer receives users' requests for buying or selling and sends negotiation requests to the negotiation control layer.

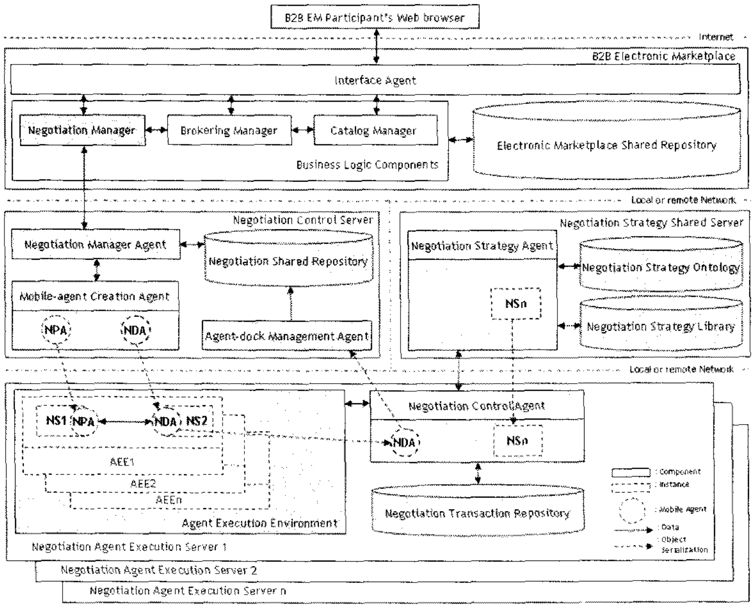
The negotiation control layer consists of a negotiation control server (NCS) and a negotiation strategy shared server (NSSS). The NCS takes charge of the whole process of negotiation and the NSSS enables negotiation agents to reuse adaptive negotiation strategies. These agent platform servers house negotiation-related agents and independent middleware, and are distributed on either local or remote networks.



〈Figure 1〉 The high-level architecture of the MANS/RNS

The negotiation execution layer accommodates several negotiation agent execution servers (NAES) in either a centralized or distributed computing environment. These agent platform servers provide a run-time environment for negotiation agent execution, and they allow several agents to be executed on the same server concurrently.

<Figure 2> illustrates the detailed architecture of the MANS/RNS at the component level (i.e., agents, objects, and databases). A conventional intermediary-mediated e-marketplace system allows users to navigate with a Web browser, a personal digital assistant, or other interfaces. This e-marketplace system consists of an interface agent (IA), a catalog manager (CM), a brokering manager (BM), and an electronic marketplace shared repository (EMSR).



〈Figure 2〉 Component-level architecture of the MANS/RNS for B2B negotiations

The IA assists users in interacting with the e-marketplace system, tracking users' behavior, and attempting to provide users with interesting items in a user-adaptive form. The CM and BM are ordinary business logic components which are parts of the B2B e-marketplace. Through the CM, market participants (buyers and sellers) post and notify products or services which they want to trade. The BM takes part in making decisions that help determine what to buy and who to buy from. The EMSR stores a typical e-marketplace's activity history, which results from the CM and BM.

The negotiation manager (NM) is one of the major concerns outlined in this article. The MANS/RNS has nine major components which are closely related with the NM, and each component is tightly integrated in order to meet the goals of achieving mobile negotiations :

- In a NCS, there are a negotiation manager agent (NMA), a negotiation shared repository (NSR), a mobile-agent creation agent (MCA), and an agent-dock management agent (AMA);
- In a NSSS, there are a negotiation strategy agent (NSA), a negotiation strategy ontology (NSO), and a negotiation strategy library (NSL);
- There are a negotiation control agent (NCA) and a negotiation transaction repository (NTR) in a NAES.

While a user sends a request for creating a negotiation agent (NA) to the NMA, he or she details highly-tailored negotiation parameters and constraints, such as negotiation issues. The NMA stores the user's request on the NSR and it is forwarded to the MCA. The MCA locates a suitable NAES, creates NAs, and launches NAs onto the NAES which was selected in the network. A NA is classified as either a negotiation provider agent (NPA) or a negotiation decider agent (NDA). A buyer (or a seller) can be an owner of either the NPA or the NDA.

At this moment, a NA can move to a NAES where the computational load is low or the connection to the NCS is less congested. An agent management system (AMS) agent—a mandatory component of the Java Agent DEvelopment (JADE) environment—helps the MCA to do this load balancing and it enables mobile agents to locate a NAES in which they reside. A traveling NA consists of a code, data, and state. The transmission of these things is handled through Java's object serialization methods and the agent system's class-loading mechanism.

The NMA registers new NAs at the NSR, monitors the progress of agents, and maintains the state of agents and system statistics. A user can see and manage mobile agents after they are registered at the repository. When an authenticated user asks about agent information, the NMA displays all of the user's registered agents.

In an agent execution environment on a NAES, the NAs communicate directly with each other and actively negotiate on the owner's behalf, even when the user is disconnected from the network. This can eventually reduce network traffic and the agents can continue to operate while network connections are down. In addition, since agents are executed on a separate server, an e-marketplace site can relieve the burden of executing a computationally-intensive negotiation strategy.

Once a negotiation agent travels to a NAES, it registers itself on the server through the NCA. The NCA manages the life cycle of an agent and it provides the mechanism to destroy agents. If a user requests to terminate an agent, the NMA receives the message from the user and sends it to the NCA. The NCA terminates the agent in question and the NMA makes the killed agent unregistered from the NSR.

Before executing negotiations, the NCA refers to the negotiation specifications of the NAs and asks the NSA to choose appropriate negotiation strategies. The NSA refers to the NSO, and then, it generates a negotiation strategy object from the template negotiation strategy classes that have been installed in advance in the NSL. Negotiation strategies are the formal descriptions of negotiation rules. They include complex computations, and are implemented by using game-theory or machine-learning techniques.

When executing negotiations, the NAs use the NTR as a safe workplace, which can store intensive calculation history. When a NA puts forth a request, the NCA records what a NPA negotiates with a NDA into the NTR. The NCA reports the final results to the NSR. A user does not need to be continuously connected to the e-marketplace site, because the NMA can send an e-mail to the user when a negotiation ends or the user can see the results through the NMA whenever he or she wants to.

3.2 Negotiation strategy

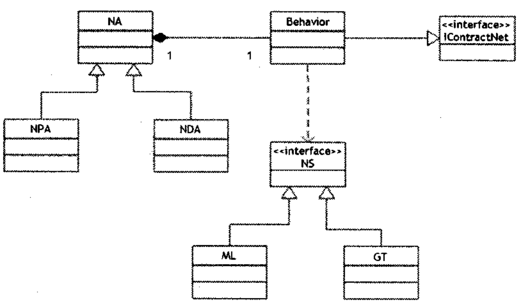
Enabling negotiation between agents allows the exchange of negotiation messages. The negotiation can be seen as a sequence of negotiation messages that are generated by the participating agents. An agent constantly receives and produces such messages. These messages are the result of intentional computations taken by the individual agents in order to achieve their goals. These intentional computations are modeled as negotiation strategies (Oliver, 1996; Jennings et al., 2001).

In order to provide a variety of negotiation strategies currently available, the MANS/RNS maintains a library of preinstalled negotiation strategies and it allows a user to choose a particular type of strategy from the library. In order to set up the library, we examined negotiation strategy literature and found that there are mainly two types of techniques that implement negotiation strategies : machine learning and game-theoretic techniques. Accordingly, the MANS/RNS is designed and developed to allow agents to negotiate with these techniques.

- Machine learning techniques. Several research papers use Genetic Algorithm (GA) or Evolutionary Algorithm as an example of a negotiation strategy (Tu et al., 2000; Choi et al., 2001; Rubenstein-Montano and Malaga, 2002; Gerding et al., 2003; Dzeng and Lin, 2004; Bădică et al., 2005). In GA, negotiation strategies can be modeled as simple sequential rules with a utility threshold. In order to enhance the strategy model, GA often employs game theory, Q-learning, or finite state machines either separately or in conjunction. Another widely used method is fuzzy constraint-based reasoning. Crisp and fuzzy relational analyses or fuzzy rule inference with simple-weighting decision are used in its variations. Bayesian learning or rule-based knowledge is also used as a negotiation strategy.

- Game-theory techniques. Game-theory techniques encompass the Adjusted-Winner (AW) procedure, Multi-Attribute Utility Function (MAUT), Pareto optimal solutions, the Kasbah model, and other scoring functions (Ueyama and Madeira, 2001; Chen et al., 2002; Karp et al., 2004; Neubert et al., 2004; Chen et al., 2005).

<Figure 3> illustrates how to define classes that encapsulate and reuse different types of strategies in Unified Modeling Language (UML), in order to construct a negotiation strategy library.



〈Figure 3〉 The pattern structure of a negotiation strategy expressed in UML notation

A negotiation agent (i.e., NPA and NDA) includes negotiation behaviors which implement the Iterated Contract Net Interaction Protocol of the Foundations of Intelligent Physical Agents (FIPA, 2002). This standard protocol allows multi-round iterative bidding. It enables negotiation agents to communicate with each other via a common interface, even though they have a variety of negotiation strategies.

The owner of the negotiation agent can specify which type of negotiation strategy should be used. A negotiation agent maintains a connection with a negotiation strategy object. The negotiation agent and negotiation strategy object interact to implement the chosen strategy. Whenever an agent generates a negotiation message, it forwards this responsibility to its strategy object.

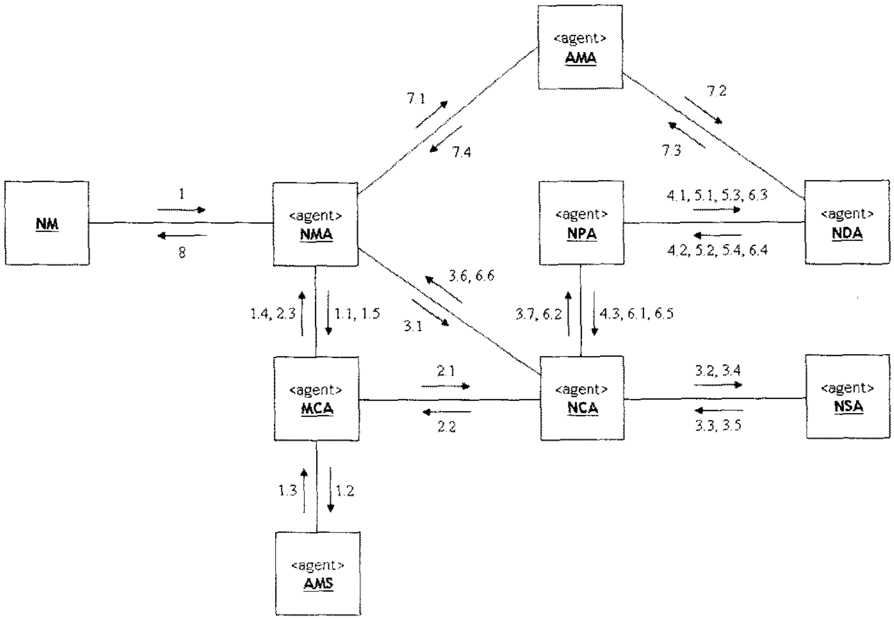
An abstract class *NS* declares an interface which is common to all supported strategies. A subclass *ML* specifies an interface that is common to its own strategies using machine learning techniques. A subclass *GT* clarifies an interface common to game-theory strategies. A negotiation agent uses these interfaces to call concrete algorithms, such as GA, fuzzy logic, or MAUT. When the algorithm is called, a NA passes all required data and parameters to the algorithm.

Hierarchies of negotiation strategy classes help to define a family of algorithms that negotiation agents can choose and reuse (Gamma et al., 1995). By encapsulating an algorithm into separate classes, it is easier to vary the algorithm dynamically independent of the negotiation agents. A user, however, must be able to understand how concrete strategies have changed before he or she selects an appropriate one. In order to reduce this overhead, the MANS/RNS implements a negotiation strategy ontology component that is responsible for managing the negotiation strategies in a modular and extensible fashion. The ontology is used to handle the registration of negotiation strategies and the notification of them to the user. It is also used to manage version control for the strategies.

3.3 Collaboration and messaging between agents

Collaboration among member agents is comprised of the dynamic interaction of agents in the negotiation system. In this article, collaboration can occur in accordance with the seven negotiation stages : Negotiation agent creation (step 1); negotiation agent migration (step 2); negotiation strategy loading (step 3); negotiation process initiation (step 4); negotiation progress (step 5); negotiation completion (step 6); and negotiation process termination (step 7).

<Figure 4> shows a collaboration diagram which is expressed in the agent unified modeling language, which explores associations among agents and the sequence of messages that are exchanged among agents during negotiation (Odell et al., 2001). The arrows represent messages and they are labeled with a dot-separated list of step numbers and sequence numbers. Messaging within the NCS occurs ten times, including two times with the AMS agent. Messaging within the NAES occurs 13 times, including iterative proposals between the NPA and NDA. Messaging takes place five times between the NCA and NAES. Messaging between



〈Figure 4〉 A collaboration diagram according to the messaging sequence among agents the NSA and NAES happens four times.

<Table 1> describes messaging details among agents. The Agent Communication Language (ACL) of the FIPA is used as the communica-

〈Table 1〉 Details of messages communicated between agents

Stage	Sequence	Sender	Receiver	FIPA ACL	Content
Step 1	1.1	NMA	MCA	request	Creation of NAs
	1.2	MCA	AMS	request	NAES information
	1.3	AMS	MCA	inform	NAES location
	1.4	MCA	NMA	inform, failure	Creation of NAs
	1.5	NMA	MCA	request-when	Migration of NAs
Step 2	2.1	MCA	NCA	query-if	Migration of NAs
	2.2	NCA	MCA	inform, failure, not-understood	Migration of NAs
	2.3	MCA	NMA	inform, failure	Migration of NAs
Step 3	3.1	NMA	NCA	request-when	Negotiation strategy loading
	3.2	NCA	NSA	inform-ref	Negotiation strategy for NPA
	3.3	NSA	NCA	inform	Negotiation strategy object
	3.4	NCA	NSA	inform-ref	Negotiation strategy for NDA
	3.5	NSA	NCA	inform	Negotiation strategy object
	3.6	NCA	NMA	inform, failure	Negotiation strategy loading
	3.7	NCA	NPA	request-when	Negotiation process initiation
Step 4	4.1	NPA	NDA	request-when	Negotiation process initiation
	4.2	NDA	NPA	inform, failure	Negotiation process initiation
	4.3	NPA	NCA	inform, failure	Negotiation process initiation
Step 5	5.1	NPA	NDA	Iterated contract net interaction protocol	Proposal
	5.2	NDA	NPA	Iterated contract net interaction protocol	Proposal
	5.3	NPA	NDA	accept-proposal, reject-proposal	Proposal
	5.4	NDA	NPA	inform, failure	Negotiation completion
Step 6	6.1	NPA	NCA	inform	Negotiation completion
	6.2	NCA	NPA	request-when	NPA destruction
	6.3	NPA	NDA	request-when	Object for negotiation results
	6.4	NDA	NPA	inform, failure	Object for negotiation results
	6.5	NPA	NCA	inform, failure	NPA destruction
	6.6	NCA	NMA	inform	Negotiation process completion
Step 7	7.1	NMA	AMA	request-when	NDA destruction
	7.2	AMA	NDA	request-when	Storing negotiation results
	7.3	NDA	AMA	inform, failure	Storing negotiation results
	7.4	AMA	NMA	inform, failure	NDA destruction

tive protocol (Wooldridge, 2002; FIPA Communicative Act Library Specification, 2002).

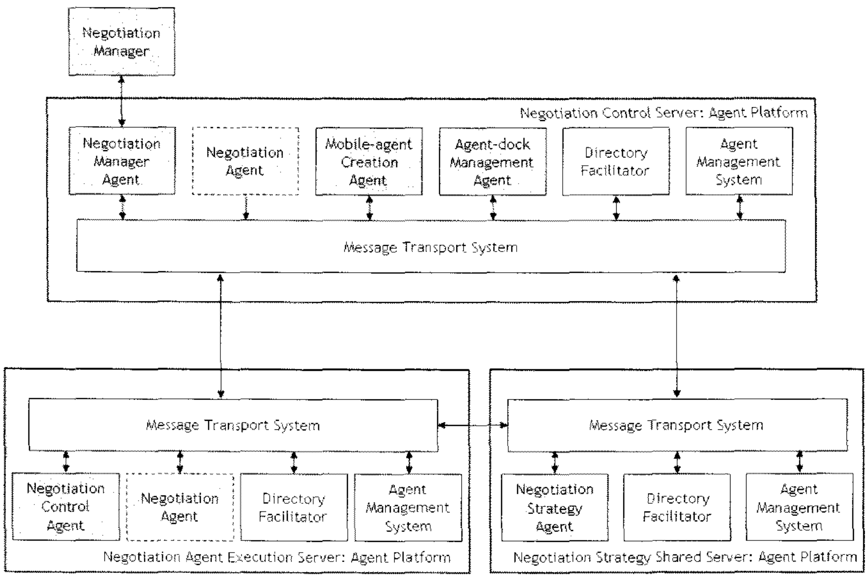
4. Implementation of the MANS/RNS

Despite some disadvantages, such as inadequate support for resource control, no protection of references, and no support for the preservation and resumption of the execution state, a mobile agent that is based on Java language has numerous advantages, including platform independence, secure execution, dynamic class loading, multi-thread programming, and object serialization.

Therefore the MANS/RNS was developed under JADE 3.0, which is fully implemented in Java language and is a framework for the development of multi-agent systems in compliance with FIPA specifications (Chmiel et al., 2004; Bădică et al., 2005). As a detailed development environment, we used FIPA ACL, JDK 1.4, Borland Enterprise Studio 7, and Rational XDE Developer for Java.

4.1 Load balancing via mobile agents in a distributed environment

Load balancing is especially important for the MANS/RNS where it is difficult to predict the number of requests that will be issued to demand negotiations and generate numerous NPAs and NDAs. The reusing of negotiation strategies increases the number of objects in negotiations and this becomes another burden of the e-marketplace system. The negotiation system, however, reduces overhead by implementing several NAESs which initiate agents on a separate server. Requests can be forwarded to another server with idle capacity, if one server starts to get congested. This leads to the more



〈Figure 5〉 Agent platform structure and inter-agent communication even distribution of processing and communication activity across a computer network.

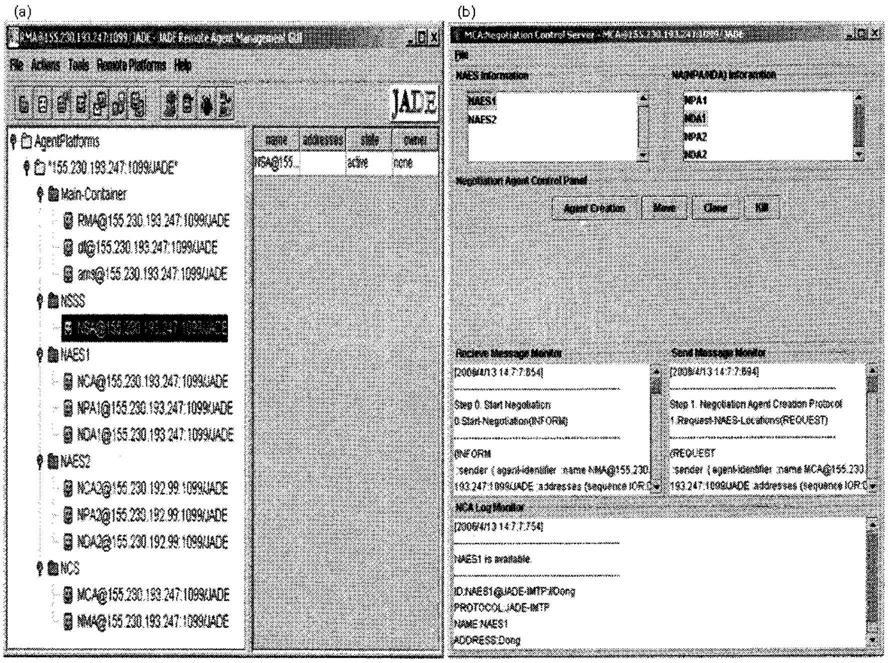
The MANS/RNS employs additional strategy in order to achieve a load-balancing scheme : mobile agents. To acquire mobile characteristics, the negotiation system should have the following structure, as shown in <Figure 5>.

The NCS, NSSS, and NAES are agent platform servers. Each of them has an AMS, a directory facilitator, and a message transport system (MTS), as defined by the FIPA. The agents of the NMA, MCA, and AMA exist in the agent platform server NCS; agent NCA is in server NAES; agent NSA is in the NSSS server. The MTS controls all exchanges within the platform and to/from remote platforms.

Under this communication infrastructure, in order to conduct load balancing, agent mobility can be realized in three ways : migration of negotiation agents (NPA and NDA) to a NAES; transmission of negotiation strategy objects to a NAES; and movement of negotiation-result objects back to the NCS.

<Figure 6> (a) illustrates a GUI of the Remote Management Agent (RMA) which is the main tool for managing JADE. <Figure 6> (b) depicts a MCA screen which is in charge of generating negotiation agents in the NCS.

The main container is an agent container that is used internally by JADE. The other agent containers, such as NSSS, NAES1, NAES2, and NCS, are connected to the main container via the remote method invocation registry, and provide a run-time environment for the execution of a set of negotiation agents. According to <Figure 6>, the NCS, NSSS, and NAES1



〈Figure 6〉 The GUI of the RMA and MCA developed in Java application form exist in the same host computer. The NAES2 resides in other host computers.

The MCA generated NPA1 and NDA1 on NAES1, and created NPA2 and NDA2 on NAES2. The followings are input parameters for generating mobile negotiation agents in this system : agent information, such as user identifier and agent identifier; negotiable issues which both parties agree on and desirable values for the desirable item; negotiation constraints such as the number of maximum trials or other termination conditions; and a type of negotiation strategy such as GA, fuzzy logic, or MAUT.

4.2 The reusing of negotiation strategies

When called upon to find an appropriate negotiation strategy, the NSA refers to the NSO component with the negotiation specifications as specified. Given the information of a strategy class, the NSA attempts to locate the class file from the NSL. The NSA creates an instance of this negotiation strategy for a negotiation agent and sends it back to the NCA.

The interaction between agents that are related with reusing of negotiation strategies can be described formally using pseudo-code notation (refer to <Table 2>) :

〈Table 2〉 Pseudo-code expressing the interaction between negotiation agents

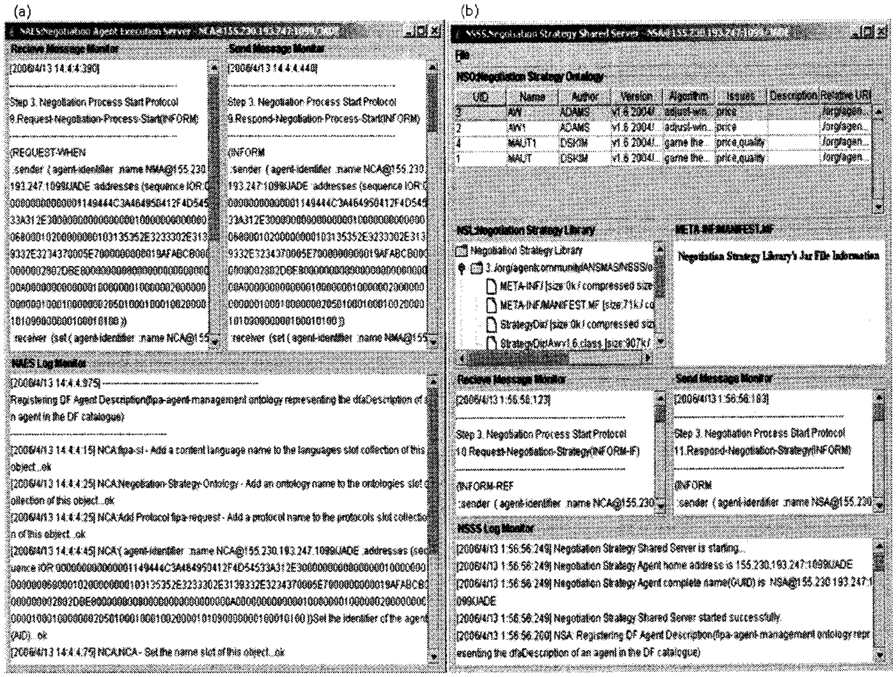
```
If migration of NPA and NDA is completed
  NMA requests NCA to start negotiation process;
    if the NMA requests
      for loop from NPA to NDA
        NCA requests a strategy for a negotiation
         agent to NSA;
      end for
      if the NCA requests negotiation strategies,
        for loop from NPA to NDA
          NSA looks up NSO;
          NSA loads up a negotiation strategy class
           from NSL;
          NSA generates a negotiation strategy object;
          NSA sends it to NCA;
        end for
        if NSA sends all strategy objects for NPA and
         NDA
          NCA informs NMA of the initiation of nego-
           tiation process;
          NCA requests NPA to initiate a negotiation;
        else
          NCA informs NMA of the failure to initiate
           negotiation process;
        end if
      end if
    end if
end if
```

<Figure 7> (a) shows a GUI that allows the NCA in NAES to provide functions to submit requests for negotiation strategies. This screen can be used to monitor all FIPA ACL messages issued from/to the NCA in order to request negotiation strategies.

<Figure 7> (b) illustrates a GUI screen that was developed to facilitate the management of negotiation strategies. The NSO contains information about negotiation strategy classes, such as strategy name, authors, version, algorithm, negotiable issues, description, and storage location. An e-marketplace provider or third parties can supply negotiation strategies in the form of Java's JAR file in advance. Each JAR file contains a Java class, which implements a negotiation strategy.

5. Comparison to other systems

To a degree, the MANS/RNS is similar to



〈Figure 7〉 GUIs showing a negotiation strategy requester (NCA) and a strategy manager (NSA)

Nomad, a famous mobile agent system having been integrated with eAuctionHouse or eMediator. Nomad used the Concordia system which was a robust framework that could develop and manage mobile agent applications. It included a Java Virtual Machine, a Concordia Server, and at least one Concordia agent. It was implemented in Java and supported mobile agents written in Java. As Nomad could have many Concordia servers, the MANS/RNS can have many agent execution environments. This characteristic ensures the extensibility of an e-marketplace which tries to meet the requirement of supporting current mobile agent systems and their execution.

The MANS/RNS, however, has several characteristics which are different from Nomad. In the Nomad, agents migrated from Concordia to the eAuctionHouse, or traveled from one agent dock to another. However, the MANS/RNS allows NAs to travel to NAESs where the NAs negotiate with each other. Agents flow in the opposite direction between servers, as compared with Nomad. This flow can lessen the burden of an e-marketplace server since a NAES executes computationally intensive negotiation strategies.

Another difference comes from a method that supports the creation of mobile agents. Nomad allowed users to program their own agents or launch template agents that had been designed and programmed in advance. However, this has some limitations. Non-programmer users are not always successful in programming their agents. In addition, the number of template agents is limited. Nomad had a limited number of bidding strategies (auction settings).

On the other hand, the MANS/RNS allows users to choose a type of pre-programmed template strategies from a negotiation strategy library. Third parties can supply template strategies and an e-marketplace intermediary can plug them into the library. This can provide the users with a variety of negotiation strategies currently available. Therefore the MANS/RNS can improve user satisfaction.

The third difference is the location of an agent database. An agent database stores information about agents, such as their creators and the negotiation terms. Each Nomad system kept its own agent database. If this is the case, users cannot monitor the progress or the state of agents when network connections between Nomad and eAuctionHouse/eMediator are down. However, the MANS/RNS maintains the database (i.e., NSR) at the NCS. Since the state of the agents is updated in real-time fashion, the users can constantly monitor the agents.

The last difference is that in Nomad, an auction database stored all of the agents' bidding history whenever a bid was made online. In the MANS/RNS, the NTR stores the intermediate outcome and the NSR keeps the final negotiation results. *Division of works*. When a user wants to see intermediate negotiation results, the NCA retrieves them from the NTR and returns them to the NMA for the user. <Table 3> summarizes the differences between MANS/RNS and Nomad.

6. Conclusion

In this article, we presented a framework and its prototype system outlining the competitive negotiation of autonomous mobile agents in a typical B2B electronic commerce scenario. The system described here offers several characteristics over a popular mobile agent system :

- The MANS/RNS supports bilateral, multi-issue negotiations while eAuctionHouse or eMediator with Nomad only supports

〈Table 3〉 The differences between MANS/RNS and Nomad

	MANS/RNS	Nomad
Agent creation	Supports bilateral, multi-issue negotiations	Supports single-item, single-unit negotiations
Agent migration	Migrates to NAES servers.	Migrates from Concordia to eAuctionHouse/eMediator, or travels from one agent dock to another.
Negotiation strategy	Pluggable negotiation strategy into the library	A fixed number of bidding strategies
Location of agent database	Negotiation Control Server	eAuctionHouse/eMediator
Bidding history	Intermediate outcomes in Negotiation Transaction Repository; final results in Negotiation Shared Repository	Auction database

single-item, single-unit auctions.

- To reduce the workload of a server running a B2B e-marketplace, the system allows the NAs to migrate from the NCS to NAESs.
- Users can choose template strategies from the NSL in order to generate their own strategies. Since the MANS/RNS allows this type of component-based development of negotiation strategies, third parties can build templates for an e-marketplace intermediary. This can improve user satisfaction since the system provides a variety of negotiation strategies.
- The strategy pattern of negotiation strategy classes makes it easy for negotiation agents to reuse a family of algorithms. By encapsulating the algorithm in separate classes, this makes it easier to modify the algorithm independent of its negotiation agent.
- By keeping an agent database at the NCS, this enables users to keep track of agents' states, even if communication between the NCS and NAES is down.
- To achieve the division of labor, the NTR stores intermediate outcomes and the NSR stores the final negotiation results.

The MANS/RNS framework can also be applied to other scenarios which need bilateral, multi-issue negotiations. This is because each component in the negotiation agent system is designed in a modular, extensible fashion so that it can be independently developed. The mobile agent system presented in this article is being enhanced to provide personalized service. User preferences about negotiations can be mined from the user negotiation history which resides in the NTR and NSR. In addition, we are working on the tuning of the MANS/RNS.

References

[1] Bădică, C., Bădiţă, A., Ganzha, M., Iordache, A., and Paprzycki, M., "Rule-based framework for automated negotiation : initial implementation", *Lecture Notes in Computer Science*, Vol. 3791, 2005, pp. 193-198.

[21] Bădică, C., Ganzha, M., Paprzycki, M., and Pîrvănescu, A., "Experimenting with a multi-agent e-commerce environment", *Lecture Notes in Computer Science*, Vol. 3606, 2005, pp. 393-402.

[3] Bartolini, C., Preist, C., and Jennings, N. R., "A software framework for automated negotiation", *Proceedings of the 3rd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems*, 2004, pp. 213-235.

[4] Baumann, J., Hohl, F., Rothermel, K., and Straßer, M., "Mole-Concepts of a mobile agent system", in Mobility : processes, computers, and agents, ACM press/Addison-Wesley Publishing, New York, NY, USA, 1999, pp. 535-554.

[5] Chen, J.-H., Chao, K.-M., Godwin, N., Reeves, C., and Smith, P., "An automated negotiation mechanism based on co-evolution and game theory", *Proceedings of the ACM symposium on Applied comput-*

ing, 2002, pp. 63-67.

[6] Chen, J.-H., Chao, K.-M., Godwin, N., and Soo, V.-W., "Combining cooperative and non-cooperative automated negotiations", *Information Systems Frontiers*, Vol. 7, No. 4-5, 2005, pp. 391-404.

[7] Chiu, C.-F., Shih, T. K., Wu, P.-Y., and Shiau, S.-H., "The design of interactive negotiation agent on the Web", *Proceedings of 22nd International Conference on Distributed Computing Systems Workshops*, 2002, pp. 176-181.

[8] Chmiel, K., Tomiak, D., Gawinecki, M., Karczmarek, P., Szymczak, M., and Paprzycki, M., "Testing the efficiency of JADE agent platform", *Proceedings of the 3rd International Symposium on Parallel and Distributed Computing*, 2004, pp. 49-57.

[9] Choi, S. P. M., Liu, J., and Chan, S.-P., "A genetic agent-based negotiation system", *Computer Networks*, Vol. 37, No. 2, 2001, pp. 195-204.

[10] Chung, M. and Honavar, V., "A negotiation model in agent-mediated electronic commerce", *Proceedings of International Symposium on Multimedia Software Engineering 2000*, pp. 403-410.

[11] Dignum, F. and Corts, U., Agent-mediated electronic commerce III, Lecture Notes in Computer Science, Vol. 2003, Springer-Verlag, Berlin, Germany, 2001.

[12] Dzeng, R.-J., and Lin, Y.-C., "Intelligent agents for supporting construction procurement negotiation", *Expert Systems with Applications*, Vol. 27, No. 1, 2004, pp. 107-119.

[13] FIPA (Foundation for Intelligent Physical Agents) Communicative Act Library Specification, URL : <http://www.fipa.org/specs/fipa00037/SC00037J.html>, 2002.

[14] FIPA (Foundation for Intelligent Physical Agents) Iterated Contract Net Interaction Protocol Specification, URL : <http://www.fipa.org/specs/fipa00030/SC00030H.html>, 2002.

[15] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., Design patterns, Addison-Wesley, MA, USA, 1995.

[16] Gerding, E., van Bragt, D., and Poutré, H. L., "Multi-issue negotiation processes by evolutionary simulation, validation and social extensions", *Computational Economics*, Vol. 22, No. 1, 2003, pp. 39-63.

[17] Gray, R. S., "Agent Tcl : A flexible and secure mobile-agent system", *Proceedings of the 4th Annual Tcl/Tk Workshop*, 1996, pp. 9-23.

[18] Guttman, R. H., Moukas, A. G., and Maes, P., "Agent-mediated electronic commerce : a survey", *The Knowledge Engineering Review*, Vol. 13, 1998, pp. 147-159.

[19] Jennings, N. R. and Bussmann, S., "Agent-based control system : why are they suited to engineering complex systems", *IEEE Control Systems*, Vol. 23, No. 3, 2003, pp. 61-73.

[20] Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., and Sierra, C., "Automated negotiation : prospects, methods and challenges", *Group Decision and Negotiation*, Vol. 10, No. 2, 2001, pp. 199-215.

[21] Johansen, D., Sudmann, N. P., and van Renesse, R., "Performance issues in TACOMA", *Proceedings of the 3rd Workshop on Mobile Object Systems in the 11th European Conference on Object-Oriented Programming*, 1997.

[22] Kang, J. Y. and Lee, E.-S., "A negotiation model in electronic commerce to reflect multiple transaction factors and learning", *Proceedings of 12th International Conference on Information Networking*, Jan. 1998, pp. 275-278.

[23] Karp, A. H., Wu, R., Chen, K.-Y., and Zhang, A., "A game tree strategy for automated negotiation", *Proceedings of the 5th ACM conference on Electronic commerce*, 2004, pp. 228-229.

[24] Kowalczyk, R., and Van Anh Bui, "FeNAs : a fuzzy e-negotiation agents system", *Proceedings of the IEEE/IAFE/INFORMS Conference on Computational Intelligence for Financial Engineering*, 2000, pp. 26-29.

[25] Lai, R. and Lin, M.-W., "Modeling of agent negotiation via fuzzy constraint processing", *Proceedings of 10th IEEE International Conference on Fuzzy Systems*, Vol. 1, 2001, pp. 445-448.

[26] Lin, F.-C. and Kuo, C.-N., "Cooperative multi-agent negotiation for electronic commerce based on mobile agents", *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2002, Vol. 3, pp. 532-537.

[27] Neubert, R., Görlitz, O., and Teich, T., "Automated negotiations of supply contracts for flexible production networks", *International Journal of Production Economics*, Vol. 89, No. 2, 2004, pp. 175-187.

[28] Odell, J. J., Parunak, H. V. D., and Bauer, B., "Representing agent interaction protocols in UML", *Lecture Notes in Computer Science*, Vol. 1957, 2001, pp. 121-140.

[29] Oliver, J. R., "On artificial agents for negotiation in electronic commerce", *Proceedings of the 29th Hawaii International Conference on System Sciences*, Vol. 4, Jan. 1996, pp. 337-346.

[30] Oliver, J. R., "A machine-learning approach to automated negotiation and prospects for electronic commerce", *Journal of Management Information Systems*, Vol. 13, No. 3, 1996, pp. 83-112.

[31] Paprzycki, M. and Abraham, A., "Agent systems today; methodological considerations", *Proceedings of the International Conference on Management of e-Commerce and e-Government*, 2003, pp. 416-421.

[32] Peine, H. and Stolpmann, T., "The architecture of the Ara platform for mobile agents", *Lecture Notes in Computer Science*, Vol. 1219, 1997, pp. 50-61.

[33] Rebstock, M., "Efficiency and flexibility of multi-attribute negotiations-the role of business object frameworks", *Proceedings of 12th International Workshop on Database and Expert Systems Applications*, 2001, pp. 742-746.

[34] Rubenstein-Montano, B., and Malaga, R. A., "A weighted sum genetic algorithm to support multiple-party multiple-objective negotiations", *IEEE Transactions on Evolu-*

tionary Computation, Vol. 6, No. 4, 2002, pp. 366-377.

[35] Sandholm, T., "eMediator : A next generation electronic commerce server", *Computational Intelligence*, Vol. 18, No. 4, 2002, pp. 656-676.

[36] Sandholm, T. and Huai, Q., "Nomad : mobile agent system for an Internet-based auction house", *IEEE Internet Computing*, Vol. 4, No. 2, 2000, pp. 80-86.

[37] Schoop, M. and Quix, C., "DOC.COM : a framework for effective negotiation support in electronic marketplaces", *Computer Networks*, Vol. 37, No. 2, 2001, pp. 153-170.

[38] Srivastava, V. and Mohapattra, P. K. J., "PLAMUN : a platform for multi-user negotiation", *Electronic Commerce Research and Applications*, Vol. 2, No. 3, 2003, pp. 339-349.

[39] Tkito, R. and Karmouch, A., "SHIPMAI : A secure and high performance mobile agent infrastructure", *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, 2000, Vol. 2, pp. 1027-1031.

[40] Tu, M. T., Wolff, E., and Lamersdorf, W., "Genetic algorithms for automated negotiations : a FSM-based application approach", *Proceedings of 11th International Workshop on Database and Expert Systems Applications*, 2000, pp. 1029-1033.

[41] Ueyama, J., and Madeira, E. R. M., "An automated negotiation model for electronic commerce", *Proceedings of the 5th International Symposium on Autonomous Decentralized Systems*, 2001. p. 29.

[42] van Bragt, D. D. B., and Poutré, J. A. L., "Why agents for automated negotiations should be adaptive", *Netnomics*, Vol. 5, No. 2, 2003, pp. 101-118.

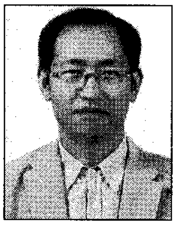
[43] Vogler, H., Spriestersbach, A., and Moschgath, M.-L., "Protecting competitive negotiation of mobile agents", *Proceedings of 7th IEEE Workshop on Future Trends of Distributed Computing Systems*, 1999, pp. 145-150.

[44] Wong, D., Paciorek, N., Walsh, T., DiCelie, J., Young, M., and Peet, B., "Concordia : An infrastructure for collaborating mobile agents", *Lecture Notes in Computer Science*, Vol. 1219, 1997, pp. 86-97.

[45] Wooldridge, M., An introduction to multiagent systems, John Wiley and Sons, 2002.

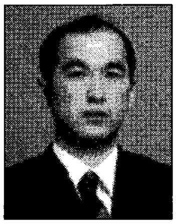
[46] Ye, Y., Liu, J., and Moukas, A., "Agents in electronic commerce", *Electronic Commerce Research*, Vol. 1, No. 1-2, 2001, pp. 9-14.

Author Profile



Sung Ho Ha

Sung Ho Ha is a professor of business administration at Kyungpook National University in Korea. His research interests include machine learning, data mining, electronic commerce, and total quality management. He received his PhD in industrial engineering from Korea Advanced Institute of Science and Technology. He is an editorial member of several SCI-indexed internationaljournals and participates in distinguished academic international conferences as a member of program committees.



Dong-Sup Kim

Dong-Sup Kim is a senior researcher and engineer at the Institute for Enterprise Content Management, IDS and Trust Co., Ltd., Korea. His research interests include intelligent agent, data mining, Enterprise 2.0, and related technologies. He received his MS in electronic commerce from Kyungpook National University.