MANS: A softbot with adaptive negotiation strategies in the B2B e-commerce

SUNG HO HA
School of Business Administration
Kyungpook National University
1370 Sangyeok-dong, Buk-gu, Daegu
South Korea
hsh@mail.knu.ac.kr http:/iiseb.knu.ac.kr

DONG-SUP KIM
Institute for Enterprise Content Management, IDS and Trust
South Korea
Dongsup.kim@gmail.com

JONG SIK JIN
School of Business Administration
Kyungpook National University
jjs@knu.ac.kr

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, e-commerce 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.

Key-Words: B2B e-commerce, Multi-agent system, Negotiation strategy, Negotiation agent

1 Introduction

Negotiation is a process of reaching an agreement on the terms of a transaction, including price and quantity, for two or more parties [3]. 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 other 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 could be 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 the key skill in distributing negotiation information [4][8].

In this article, we present MANS, a Mobile Agent Negotiation System with reusable negotiation strategies that allow agents to dynamically embody a user's favorite negotiation strategy which can be preinstalled as a component within the system. The MANS 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 has a mechanism which enables negotiating agents to combine pluggable strategy components.

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.

2 Conceptual Design and Architecture of MANS

2.1 Conceptual design

The MANS is organized into a four-layer structure (refer to Fig. 1): 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).

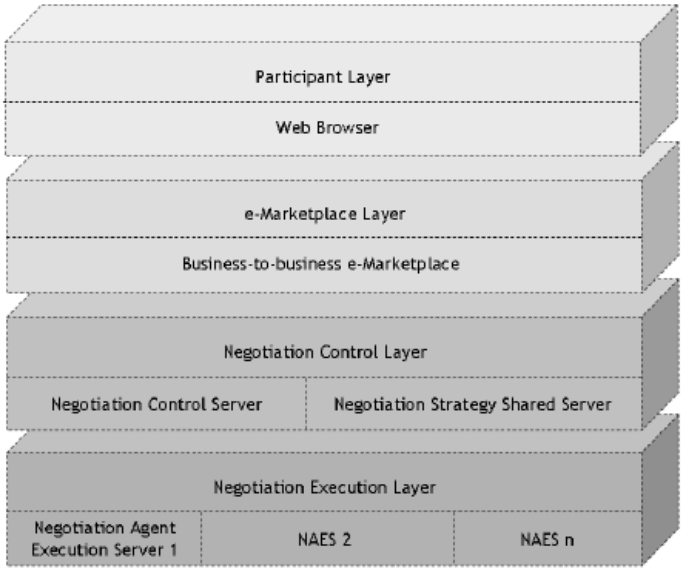

Fig. 1. Conceptual design of MANS

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.

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.

2.2 Architecture of MANS

Fig. 2 illustrates the detailed architecture of the MANS at the component level (i.e., agents, objects, and databases). This e-marketplace system consists of an interface agent (IA), a catalog manager (CM), a brokering manager (BM), a negotiation manager (NM), and an electronic marketplace shared repository (EMSR).

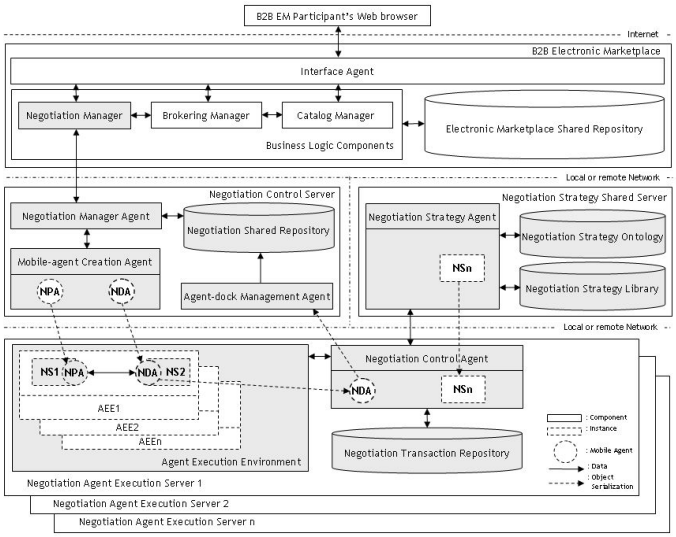

Fig. 2. Architecture of MANS

The NM is one of the major concerns outlined in this article. The MANS 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.

2.3 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 [4-5]. In order to provide a variety of negotiation strategies currently available, the MANS maintains a library of preinstalled negotiation strategies and it allows a user to choose a particular type of strategy from the library. MANS is designed and developed to allow agents to negotiate with these techniques:

- Machine learning techniques: Several research papers use Genetic Algorithm or Evolutionary Algorithm as the examples of a negotiation strategy;
- Game-theoretic techniques: Game-theory techniques encompass the Adjusted-Winner procedure, Multi-Attribute Utility Function, Pareto optimal solutions, the Kasbah model, and other scoring functions.

3 Implementation of MANS

The MANS 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 [1-2]. As a detailed development environment, we used FIPA ACL, JDK1.4, Borland Enterprise Studio 7, and Rational XDE Developer for Java.

3.1 Mobile agents in a distributed environment

The reusing of negotiation strategies increases the number of objects in negotiations and this becomes another burden of the e-marketplace system. Therefore load balancing was especially important for the MANS where it was difficult to predict the number of requests that were issued to demand negotiations and generate numerous negotiation provider agents (NPAs) and negotiation decider agents (NDAs).

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

Fig. 3 illustrates a GUI of the Remote Management Agent (RMA) which was the main tool for managing JADE. And it depicts a MCA screen which was in charge of generating negotiation agents in the NCS. The NCS, NSSS, and NAES1 existed in the same host computer. The NAES2 resided in other host computers.

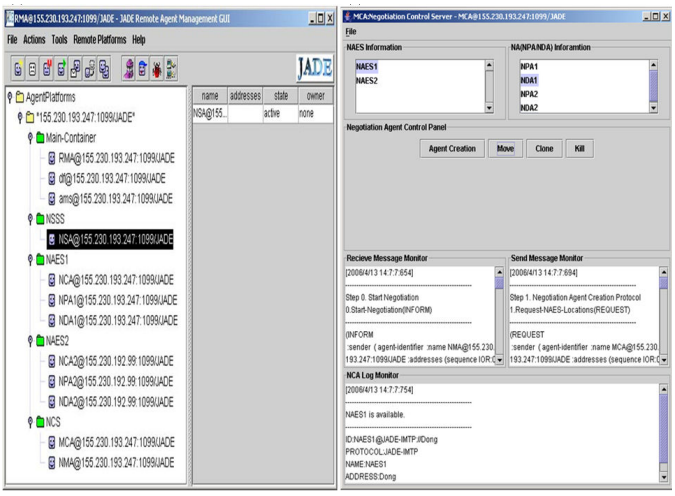

Fig. 3. GUI screens of the RMA and MCA

3.2 Reusing of negotiation strategies

Whenever called upon to find an appropriate negotiation strategy, the NSA referred to the NSO component with the negotiation specifications as specified. Given the information of a strategy class, the NSA attempted to locate the class file from the NSL. The NSA created an instance of this negotiation strategy for a negotiation agent and sent it back to the NCA. The interaction between agents that were related with reusing of negotiation strategies can be described formally using pseudo-code notation (refer to Table 1).

Table 1. 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 negotiation 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
```

The left screen in Fig. 4 shows a GUI that allowed the NCA in NAES to submit requests for negotiation strategies. This screen could be used to monitor all FIPA ACL messages issued from/to the NCA in order to request negotiation strategies to meet the requirement of supporting current mobile agent systems and their execution. The right screen illustrates a GUI that was developed to facilitate the management of negotiation strategies. The NSO contained information about negotiation strategy classes, such as strategy name, authors, version, algorithm, negotiable issues, description, and storage location.

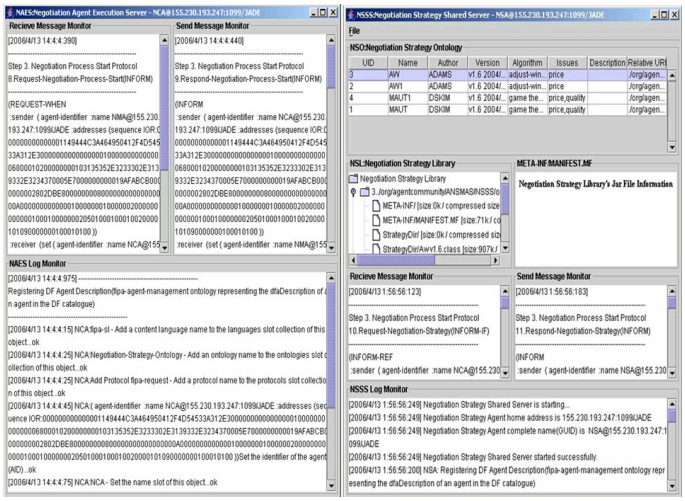

Fig. 4. GUI screens, showing a negotiation strategy requester and a strategy manager

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.

4 Comparison with Other Systems

To a degree, the MANS is similar to 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 [6-7]. 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 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.

However, this system 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 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. Table 2 summarizes the differences between MANS and Nomad.

5 Conclusion

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

- The MANS supports bilateral, multi-issue negotiations while eAuctionHouse or eMediator with Nomad only supports single-item, single-unit auctions;
- Users can choose template strategies from the NSL in order to generate their own strategies;
- The strategy pattern of negotiation strategy classes makes it easy for negotiation agents to reuse a family of algorithms. Encapsulating the algorithm in separate classes makes it easier to modify the algorithm independent of its negotiation agent;
- Keeping an agent database at the NCS enables users to keep track of agents' state, even if communication between the NCS and NAES is down;
- To reduce the workload of a server running a B2B e-marketplace, the system allows the negotiation agents to migrate from the NCS to NAESs;
- To achieve the division of labor, the NTR stores intermediate outcomes and the NSR stores the final negotiation results.

The mobile agent system presented in this article is being enhanced to provide personalized services. User preferences about negotiations could 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.

Table 2. The differences between MANS and Nomad

	MANS	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 strategies into the library	A fixed number of bidding strategies
Location of agent database	NCS	eAuctionHouse/eMediator
Bidding history	Intermediate outcomes in Negotiation Transaction Repository; final results in Negotiation Shared Repository	Auction database

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

[8] Ye, Y., Liu, J., and Moukas, A., "Agents in electronic commerce", *Electronic Commerce Research*, Vol. 1, No. 12, 2001, pp.9-14.