A FULLY-DISTRIBUTED, MULTIAGENT APPROACH TO NEGOTIATION IN MOBILE AD-HOC NETWORKS

Iván Marsá Maestre, Juan R. Velasco, Miguel A. López, and Andrés Navarro
Departamento de Automática, Universidad de Alcalá
Edificio Politécnico, Ctra N-II Km. 31,600
28871 Alcalá de Henares, Madrid, SPAIN
{ivmarsa, juanra, miguellop, andres}@aut.uah.es

ABSTRACT

This paper presents an interaction protocol intended to be used in distributed negotiation problems using software agents, which could be applied to multi-agent systems deployed over Personal Digital Assistants (PDAs) connected via wireless networks. We are especially interested in semi-competitive scenarios, where each agent in the system acts on behalf of a user, trying to maximize its user preferences while pursuing a common agreement. In these conditions, and especially if we are dealing with open and dynamic environments like mobile ad-hoc networks, the goals and attitudes of software agents cannot be guaranteed. Taking this into account we propose a protocol where interaction among agents is done in a fully-distributed manner, so that no user can have negotiation privileges over the others.

KEYWORDS

Fully-distributed negotiation, interaction protocol, mobile ad-hoc networks, multiagent systems.

1. INTRODUCTION

Our research focuses on using agents to automate routine tasks usually performed by users, in order to provide them with both more comfort and efficiency at home, in organizations and institutions. We are especially interested in integrating multi agent systems into mobile personal handheld devices, which may be used to authenticate users, store their preferences and host any personal agents they need.

In the context of a smart environment, for example, a PDA can be used both to identify and locate the user inside a building and to provide the adequate interfaces for the services available at each location. Also, ad-hoc networks have been revealed as a very suitable alternative for providing the necessary network infraestructure, due mainly to their ease of deployment at a reasonably low cost and to the inherent flexibility of their network topology, which allows easy insertion and removal of users and devices.

With the addition of mobile handheld devices to represent and locate users in the smart environment or to provide interfaces to the services available, the network infraestructure becomes a mobile ad-hoc network, or *manet*. Other applications involving mobile handheld devices, such as communication among users in airports, conferences or classrooms, can rely on ad-hoc networks to provide connectivity between its users.

Given the open and dynamic nature of *manets*, in applications that involve negotiation among users with different interests, we believe that trust in other user attitudes should not be assumed. If a negotiation schema gives more control to a certain user, other users cannot have any guarantee of that user not taking advantage of that. Such a guarantee of impartiality in the solution can be achieved via transparent communication among users and decentralised decision making. Taking this into account, we propose a fully-distributed interaction protocol schema, which facilitates the development of distributed negotiation strategies where each agent defends the interests of its user, and where no user has negotiation privileges over the others.

The rest of this paper is organized as follows. Sections 2 and 3 recall the most relevant concepts our research is based on. Section 4 outlines a fully-distributed interaction protocol, which constitutes the basis of our approach. Section 5 describes our implementation of a system based on the proposed interaction schema, outlines the scenarios set up to test it, and presents the results yielded by the tests made. The last section summarizes our main contributions and sheds light on some future research.

2. DISTRIBUTED NEGOTIATION

Negotiation research encompasses three different topics: the set of rules which govern the interaction, the range of issues over which agreement must be reached, and the agent decision making models (Jennings et al, 2000). In particular, in collaboration or competence situations among agents we need to define an interaction mechanism -a protocol- that allows agent to solve their conflicts and reach a cooperative agreement. That's what is known as negotiation protocols (Sen, S., Durfee, E., 1998). For a given negotiation problem in a multi-agent system, there may be multiple strategies based on different protocols. Such protocols may be classified according to diverse criteria. According to the degree of distribution, we can divide negotiation protocols into three categories:

Centralised negotiation protocols. There is only one agent specialized in negotiation, which provides the negotiation services to all other agents. Usually, the communication model will be centralized as well, that is, during negotiation each client agent will communicate with the server, but there is generally no need for communication between client agents. The key advantage of centralized negotiation is its effectiveness. Since the conflicts of interests and attitudes are solved by the server agent, if a solution exists, agreement is guaranteed. However, the model presents several concerns, such as fail tolerance, scalability, privacy protection and the trust issues raised by delegating decision making to a central entity.

Partially-distributed negotiation protocols. In this model all agents have the same negotiation capabilities. However, when a negotiation process starts, the negotiation tasks are centralized in one of the participant agents, usually the initiator of the process, which is called the host of the negotiation (Sen, S., 1997)., will communicate in a centralized manner with all the other participants, will evaluate the information provided by them, and will finally take a decision based on that information. This approach has the same advantages as the centralized model, and solves most scalability and fault-tolerance problems, as the host is different for each interaction. However, since all agents need to have negotiation capabilities, it increases software complexity.

Fully-distributed negotiation protocols. In this model agent communication and data storage are fully distributed. Participants interchange information transparently, by multicasting. Each agent decides what information it gives to the system, but that information is shared by all participants. Similarly, the final outcome of the negotiation is decided by each and every agent, so there is a full distribution. The main drawback of this model is its low efficiency when compared to centralized and partially-distributed negotiation as each message must be sent to all participants, which increases network resources consumption[1]. On the other hand, distribution makes this model as scalable and fault-tolerant as the previous one. Furthermore, as the final outcome of the negotiation is decided in a distributed manner, there is no need to grant privileges on decision making to any participant, so trust concerns are significantly lowered. An example of an application of a fully-distributed negotiation approach can be seen in (Wang, K., 2003).

3. MOBILE AD-HOC NETWORKS

A Mobile Ad-hoc Network, also called *manet* (Corson, S., and Macker, J., 1999), comprises a set of mobile, autonomous nodes, which are interconnected using wireless links. In a *manet*, there is no fixed network infrastructure, and network management is fully decentralised. Distant nodes can communicate using multihop paths, where intermediate nodes cooperate in relaying packets in order to deliver them to the destination nodes. Thus, mobility is supported without the need for any fixed infrastructure.

One of the key advantages of *manets* is their ease of deployment, which make them specially suitable for applications such as communications on battlefields or for providing network infrastructure in disaster recovery actions, as well as for sensor networks and smart environments (Akyildiz, I.F. et al, 2002). Other applications may include information sharing and communication among personal handheld devices (cell phones, PDAs) in airports, conferences or classrooms. The work presented here is mainly focused on this kind of device, so restrictions such as bandwith availability, computing capacity and battery power must be

---

[1] Later, we will see a way to reduce this impact over network resources for applications running in wireless ad-hoc networks, such as the architecture for PDAs that we are proposing.

taken into account. Thus, communication protocols must be kept simple, and must have lightweight computational, bandwith and information storage needs (Mohapatra et al, 2005).

Another key issue in *manet* environments involving personal devices is trust. Due to the infraestructureless nature of ad-hoc networks, it is very difficult to establish trust relationships among the nodes. Thus, assumptions about the benevolence of nodes during a negotiation should not be made. There are many open lines of research about *manet* security issues (W. Lou and Y.Fang, 2004) (Kong, J. Et al, 2001), yielding promising results. However, even if the existing and future proposals can provide a certain degree of authentication, confidentiality and integrity assurance at network and transport levels, trust at application level in open environments will remain an issue.

4. THE COUNCIL INTERACTION PROTOCOL

The main contribution of our research is to use a communication schema that facilitates the development of fully distributed negotiation strategies, which are not more efficient, but more adequate to environments where trust in other participants should not be assumed, such as competitive negotiation over mobile ad-hoc networks. Our research scenario is a group of agents trying to solve a certain negotiation problem. We assume that group has a set of public global goals, known and pursued by every agent, and that each agent also pursues its user goals. We assume also that no agent has special privileges over the other agents during the negotiation. There is an initiator who describes the problem to the other agents, but once the negotiation has started the protocol treats each participant in the same way. The final outcome of the negotiation will be decided in a distributed manner. Furthermore, information interchange among agents will be minimized, thus enforcing privacy in the system.

An intuitive solution to the problem is that the initiator sends a call for proposals to all other participants, and that each participant is allowed to respond with zero, one or more proposals, which will be sent to all the others. In this way, each participant may judge each issued proposal, and the judgement is also sent to all participants. By ensuring all participants know the opinion of all others about all proposals, it is possible to reach an agreement in a fully distributed manner, without the need for a privileged participant taking the final decision. This approach, compared to the traditional centralized and partially-distributed ones, provide a better guarantee of an impartial solution, as the participants do not need to trust an external negotiation entity. Furthermore, as the only information interchanged is inside the proposals, participants may control how much information they give to other agents about their users, which leads to privacy enhancement.

We have developed a new interaction protocol (IP) which addresses the communication model proposed in the above discussion. One of the key points in the design of the protocol has been to create a generic IP that may be used to solve different negotiation problems where full distribution is needed. The result of our research is the Council protocol, that we describe in this section. It has been designed to comply with the specifications of the Foundation for Intelligent Physical Agents (FIPA), and therefor it has been described in an analogous manner to that used by FIPA to describe its own IPs.

4.1 The Council Protocol Scenario

The Council protocol is proposed as a solution to distributed negotiation and decision problems where the following circumstances are met:

1. Regarding the negotiation problem, there is a set of public common goals G that all participants desire to meet, and a set of private goals G_i for each participant.
2. At least one subset of the participants is able to create proposals representing possible solutions to the problem.
3. At least one subset of the participants is able to make value judgements about the solutions proposed by other agents.
4. At least one subset of the participants is able to deduce a valid solution to the problem from the proposals received and the judgements made by the participants about them.
5. If there are several solutions that satisfy the conditions for agreement –e.g. to be judged positively by at least 50 per cent of the participant plus one-, there must be selection criteria –normally in the

set of common goals- that allow to decide univocally which one is selected. It is desirable that those criteria are such that no further communication among agents is required.

The protocol has been named *Council* to represent the philosophy of the approach: the controlled information interchange among equals to reach an agreement about the solution to a problem.

4.2 Conceptual Example of an Interaction Using the Council Protocol

To help to understand the above conditions and the formal description presented below, an example of an interaction among four agents is provided here. The problem under discussion is very simple: to decide the starting time of a certain event. For the example we assume that the final decision criterion is global agreement, that is, a solution is finally accepted if and only if all participants have judged it as valid. The following is the step-by-step evolution of the interaction.

1. Agent A describes the problem and issues a call for proposals to all participants, including itself. From now on, A will be treated as any other participant -Fig. 1(a)-.
2. D proposes to all other participants the starting time "09:00 GMT" as a possible solution to the problem -Fig. 1(b)-.
3. A and B accept that proposal -Fig. 1(c)-. Acceptance is represented using dashed lines.
4. B issues a new proposal -"10:00 GMT"-. C rejects D's proposal -Fig. 1(d)-. Rejection is represented using dotted lines.
5. A issues a new proposal -"11:00 GMT"-. D accepts it -Fig. 1(e)-.
6. B and C accept A's proposal -Fig. 1(f)-.

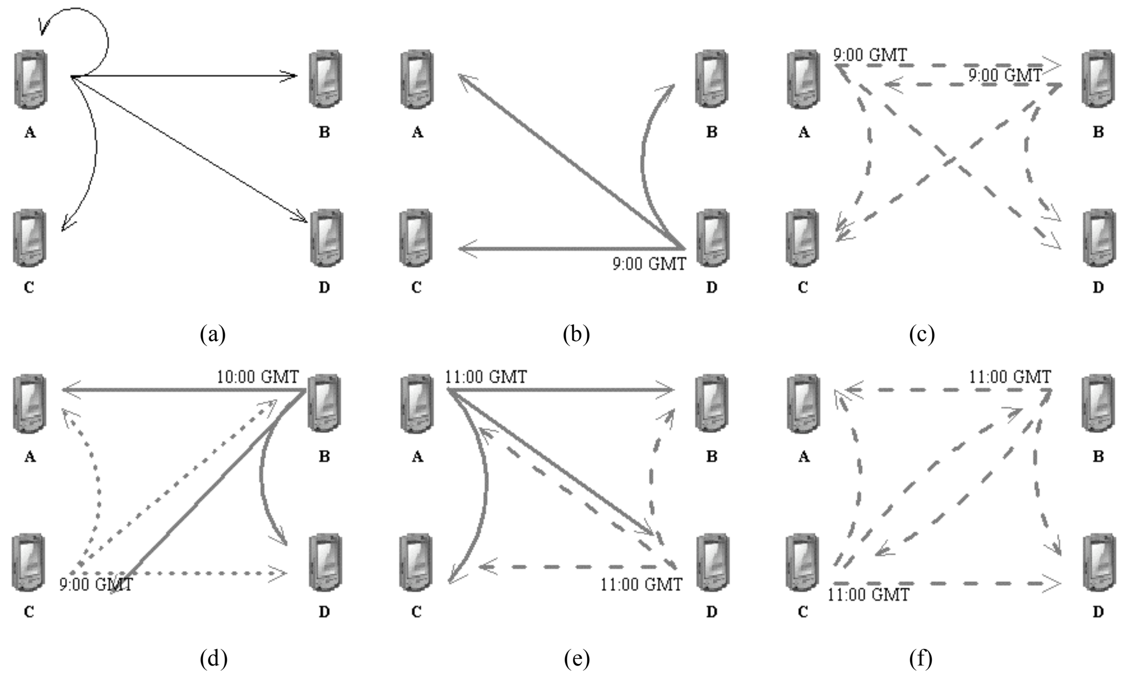


Fig. 1. Example of an interaction using the Council protocol

At this point, the solution "11:00 GMT" has been accepted by all participants -note that A, being the proposer of this solution, has implicitly accepted it-. Therefore, the interaction can be finished without any further message interchange, as all participants know the final solution. However, there is another ongoing proposal -"10:00 GMT"- and, for certain kind of problems, it may be more adequate to wait until the proposal is resolved -that is, until it has been judged by all participants- before concluding the interaction. Both cases are supported by the protocol. If multiple valid solutions are allowed, and if no further information interchange is desired before closing the interaction, the system needs a criterion that allows it to decide among them without doubt. For example, the set of common goals *G* may establish that solutions where the event starts earlier in time are preferable. In this way, if both “9:00 GMT” and “10:00 GMT” solutions have been globally accepted, the first one would be selected by all agents without needing any further information interchange.

4.3 Formal representation of the protocol

Figure 2 shows the sequence diagram of the protocol flow, using the same representation that FIPA uses to define their interaction protocols, which is based on extensions to UML1.x (Odell, J. et al, 2001).

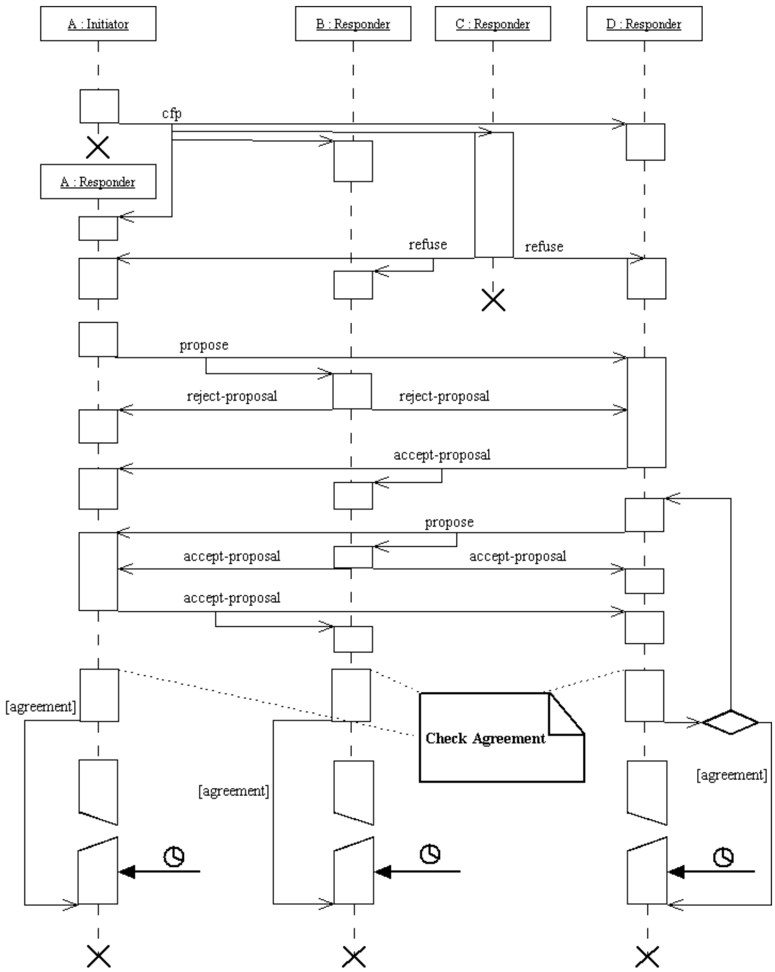


Fig. 2. Sequence diagram of the Council Interaction Protocol

4.4 Explanation of the Protocol Flow

The initiator solicits proposals from the participants by issuing a call for proposals -`cfp`- act (FIPA, 2002), which specifies the problem under discussion, as well as any conditions the initiator is placing upon the solution to the problem. It also specifies the deadline for the resolution of the interaction. All participants - that is, all agents receiving the call for proposal, including the initiator- can then issue proposals, refuse to take part in the interaction, or do neither -simply wait for the proposals of other participants-. In Fig. 2, three agents reply to the initiation message. Two of them issue a proposal by sending a `propose` act, and the other abandons the interaction sending a `refuse` act. Note that proposal and refusal messages are sent to all participants in the interaction.

Proposals include the conditions that make possible the solution to the problem described by the initiator, such as an available time interval or some quality-of-service parameters, for example, depending on the problem under consideration. Each participant evaluates each proposal upon arrival, and sends a response to all other participants. This response will be an `accept-proposal` act if the agent accepts the proposal or a `reject-proposal` act if it rejects it. Note that no further message is sent to those agents that have refused to take part in the interaction.

The check for the agreement conditions is performed whenever a message related to a proposal -that is, a `propose`, `accept-proposal`, or `reject-proposal` message- is sent or received. If there is no agreement and the deadline has not yet expired, each participant may decide to issue a new proposal to keep the interaction moving forward. The methods used to decide if there is an agreement and if a new proposal must be issued are not defined by the protocol itself, and may change in each final implementation. In Section 5, we describe our test implementation and we outline the application-dependant criteria and rules we have used.

If an agreement is reached, the interaction finishes with success. If the deadline expires without agreement or the participants decide the agreement is unreachable -for any reason imposed by the particular implementation-, the interaction terminates and a failure result is returned. Both possible conditions are detected in a distributed manner, without needing any participant to inform the others of the final result.

4.5 Exceptions over the protocol flow

At any point in the interaction protocol, the receiver of a communication can inform the sender that it did not understand what was communicated. This is accomplished by returning a `not-understood` message. As such, Fig. 2 does not depict a `not-understood` communication as it can occur at any point in the IP.

At any point in the IP, the initiator may cancel the interaction protocol by sending a `cancel` act identified by the `conversation-id` parameter associated to the canceled interaction-. The semantics of cancel should roughly be interpreted as meaning that the initiator is no longer interested in continuing the interaction.

The protocol flow described in this paper is only a model of the interaction. It does not intend to specify all cases that might occur in actual agent negotiation. Issues such as the effects of canceling actions, asynchrony, abnormal termination of the protocol in one or several agents, nested protocols, and the like, should be addressed for each specific application. We provide a basic outline of how we have addressed this issues for our test implementation in the following section.

5. TEST IMPLEMENTATION

We want to ensure the portability of the system to different computer architectures and its interoperability with other agent-based systems. Taking this into account, the system has been developed over the open-source agent platform JADE (Java Agent DEvelopment framework) (JADE).

Although our aim is to provide a fully-distributed interaction system for PDAs, the test implementation has been deployed over PCs. Using PCs allows to use all the debugging and analysis tools available with JADE and Java development environments, so that message interchange can be adequately monitored and logged, and appropriate statistical information can be generated. At the time of writing, we are working on an implementation of the system over a HP5550 device.

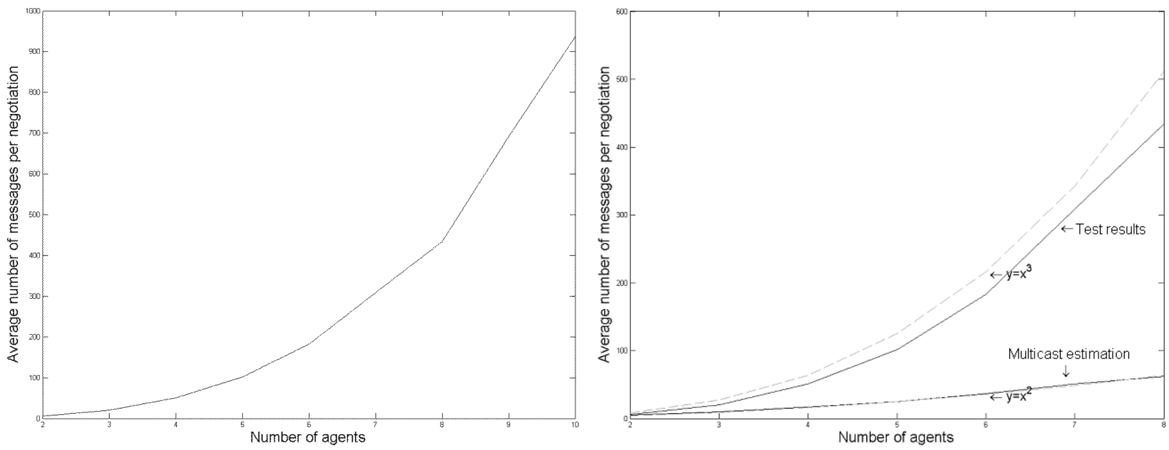
As it has been stated before, the goal of this stage of our research was the interaction protocol, and not the application-specific, heuristic algorithms for proposal generation, proposal judgement, and agreement. However, to test the interaction model it was necessary to establish the attitude of the agents. To be able to evaluate the protocol rather than the algorithms, a simple application scenario and decision-making model was used for testing. It was based on a classical distributed negotiation problem: the scheduling of a meeting with two or more participants. The test implementation model can be summarised as follows:

1. Each agent was loaded a randomly generated calendar. Calendars were generated with calendar occupation ratios averaging 80% and intersections among free time slots of the calendars of different users averaging 10%.

2. An agent may be tagged as essential by the initiator of a negotiation, with finite probability. If an essential agent refuses to take part in the negotiation, the process yields a failure result.
3. Once a negotiation has started, each participant waits a random time before issuing a proposal.
4. Each proposal consists basically of a time slot where the meeting can take place.
5. When an agent receives a proposal, it is immediately judged. Judgement of a proposal is binary: an agent accept a proposal if the proposed slot is not occupied by another appointment in its calendar, otherwise it rejects the proposal.
6. We consider there is an agreement if any proposal has been accepted by all participants and all proposals are closed -that is, all issued proposals have been judged by all participants-.
7. If all proposals are closed and there is no agreement, each agent waits a random time and issues a new proposal.
8. If there are multiple globally accepted proposals, the agents select the one which causes the meeting to take place earlier in time.

With this negotiation policy, several test scenarios were created, adjusting the number of users and the number of simultaneous negotiations. We performed tests for up to ten users, and for up to five concurrent negotiations. The system performs according to the specifications of the protocol, responding successfully to meeting scheduling requests, leading to an agreement in a reasonable time –mean time to agreement of 37 seconds-, assuming the agreement is possible. It reacts quickly to refusal events and forced communication failures, canceling the interaction only if these events involve essential users. It allows negotiation about several meetings concurrently, and those negotiations can be initiated by the same agent or by different ones.

Figure 3(a) shows there is a cubic dependency between the number of messages produced during negotiations and the number of agents involved, yielding average numbers of messages considerably higher than those obtained when using centralized or partially-distributed solutions. This is mainly due to how the JADE platform handles 1:N messages, as it issues a separate message for each receiver. One of our future lines of work is an extension of JADE that takes advantage of the diffusion capabilities of wireless networks. Fig. 3(b) shows an estimation of this reduction, comparing the test results to the calculated results for a multicast environment. The dependency with the number of agents involved will be quadratic, thus drastically reducing the network load.



(a) (b)

Fig. 3. Average number of messages per negotiation. (a) Test results. (b) Test results compared to multicast estimation .

6. CONCLUSION

Many effective multi-agent systems rely on interaction strategies. The evaluation of these interaction strategies depends on the scenarios where they are supposed to be working. For each specific application, there is a most suitable approach to negotiation, which yields optimal results or solves some important concerns the other mechanisms do not address. When dealing with negotiation among PDA users in an ad-hoc network, there is a small-to-medium number of users in an open, semi-competitive environment. Those users will probably have different goals and priorities about the object of the negotiation. An interaction mechanism was needed where no user had privileges over other users in the negotiation -that is, that the protocol was symmetric-. One solution to this is a fully-distributed negotiation strategy. Even if such a strategy is used, if the underlaying interaction schema does not provide at least the same degree of distribution, a user could be in a privileged position to manipulate information to his own advantage. Thus a fully-distributed interaction protocol is also needed, which is what this paper provides. The concept has some similarities with a blackboard communication schema (Paderewski-Rodriguez, P. Et al, 2003), but with a key difference: in the Council protocol the "blackboard" is distributed, so malicious manipulation of the information presented to participants is harder. For semi-competitive scenarios and open environments such as mobile ad-hoc networks, this is a key advantage.

Our tests over the implemented system show that this interaction protocol can be used effectively to solve distributed negotiation problems. The effectiveness and the efficiency of the negotiation process depend on the high-level, heuristic negotiation algorithms that are used on top of the protocol. Our research will focus now on this field, trying to find algorithms which guarantee a certain degree of optimality for the produced solutions and also deal with the possibility of having agents with different -and possibly conflicting-decision-making models. We are also working in an extension of JADE that takes advantage of the diffusion capabilities of wireless networks, which we hope will significantly reduce the network load, as stated before. Also, security issues of the system should be carefully studied. Finally, we are exploring other problems that could be addressed using fully-distributed negotiation, with an special interest in its application to smart environments.

ACKNOWLEDGEMENT

This work has been supported by the Ministerio de Ciencia y Tecnología grant MCYT-TIC2003-09192-C11-05, and by the Universidad de Alcalá grant UAH-PI-2003/001.

REFERENCES

Akyildiz, I.F. et al, 2002. Wireless Sensor Networks: A Survey. *In Computer Networks* Vol. 38, pp 393-422.

Corson, S., and Macker, J., 1999. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, IETF RFC 2501.

FIPA Homepage. http://www.fipa.org

FIPA, 2002. Fipa Comunicative Act Library Specification. Available at http://www.fipa.org/specs/fipa00037/

JADE Homepage. http://jade.cselt.it

Jennings et al, 2000. Automated Negotiation. *In Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems (PAAM-2000),* pp 23-30

Kong, J. Et al, 2001. Providing Robust and ubiquitous security support for manet. *In Proceeding of the 9th IEEE International Conference on Network Protocols (ICNP)*, pp. 251-260.

Lou, W. and Fang, Y., 2004. A Survey of Wireless Security in Mobile Ad-Hoc Networks: Challenges and Available Solutions. *Ad-Hoc Wireless Networking,* Cheng et al. (Eds.). Kluwer Academic Publishing, USA

Mohapatra et al, 2005. Group Communications in Mobile Ad Hoc Networks. *In IEEE Computer*, Vol.37,No. 2, pp 52-59.

Odell, J. et al, 2001. Representing agent interaction protocols in UML. *In Agent-Oriented Software Engineering*, pp. 121-140. Ciancarini, P. and Wooldrigge, M. (Eds.). Springer, Berlin.

Paderewski-Rodriguez, P. Et al, 2003. An architecture for dynamic and evolving cooperative software agents. *In Computer Standards & Interfaces,* Vol. 25, No. 3,pp 261-269

Sen, S., 1997. An automated distributed meeting scheduler. *In IEEE Expert*, Vol 12, No. 4, pp 41-45

Sen, S., Durfee, E., 1998. A formal study of distributed meeting scheduling. *In Group Decision and Negotiation*, Vol. 7, No. 3, pp. 267-289.

Wang, K., 2003. Negotiation-based multi-stage production control using distributed shortest path algorithm. *In International Journal of Computer Integrated Manufacturing*, Vol. 16, No. 1, pp. 38-47