Meeting Scheduling with Multi Agent Systems: Design and Implementation

ISMAIL OZGUR DEMIREL, NADIA ERDOGAN
Computer Sciences, Informatics Institute
Istanbul Technical University
Maslak-34469, Istanbul
TURKEY
{demirel, nerdogan}+@itu.edu.tr

Abstract: - In this paper, we present the design and implementation of a distributed multi agent meeting scheduling system. The system includes two types of agents: personal agent, one for each user, to negotiate on behalf of its owner, and a location agent to arrange suitable meeting rooms for the meetings. Personal agents have their own calendar and preferences. Invitees are classified as either important, without whom a meeting cannot be held or regular, whose absence does not prevent a meeting from being held. An agent has the ability to produce a counter proposal when it cannot accept a proposed date due to a conflict with its calendar, thus reducing negotiation time. Both types of agents can carry out several negotiations concurrently. We've made experiments to show how scheduling time and efficiency of the negotiation protocol vary with different calendar densities and different numbers of meetings.

Key-Words: - multi agent system, meeting scheduling, JADE, negotiation, distributed search

1 Introduction

In business life, meetings are frequently held to share ideas and to keep coordination. Meeting scheduling is a time consuming routine task that can sometimes become very boring. Most of the time, people need to schedule numerous meetings at the same time, taking into account several constraints, such as a suitable time slot, a suitable location, and/or a certain number of attendees. The process of searching for a commonly available time, place, etc., can be frustrating and lead to sub-optimal solutions in the presence of several meetings being scheduled concurrently. Therefore, automating meeting scheduling can save time and effort on the part of participants and also may lead to more efficient schedules.

Multi agent systems provide suitable solutions for scheduling because scheduling is mostly an iterative task that can result in waste of time [1]. The distributed nature of meeting scheduling, mainly involving a distributed calendar search, also matches well with agent based distributed computing model. Moreover, goal oriented, autonomous agents, continuous in time, fit perfectly into the process as they can decide how the task is to be achieved on behalf of their user and they can perform the necessary set of actions for scheduling [2]. As the agents handle interactions with other agents, users are not involved in negotiations and need not monitor the scheduling process. Use of multi agents also reduces the classical client/server bottleneck problem.

Various approaches have been proposed to implement meeting scheduling systems. Sen and Durfee [1] claim that meeting scheduling as a distributed search process and they give a formal description for the meeting scheduling. Scott and others [3] discuss meeting scheduling in large and geographically distributed issue. They use a facilitator agent which is responsible for responding to meeting requests from any initiating meeting agent. In addition, the paper concentrates on privacy, scalability, user control, reasonable meeting times and trust issues. Crawford and Velose [4] have implemented a negotiation system strategy that sends all of available time intervals to all agents. They describe an "open negotiator" to show open calendar negotiation.

In our approach, agents negotiate by having one agent propose a meeting, which the other agents accept or reject, based on whether or not it fits their own schedules. There are two types of agents, personal agent and location agent. Each personal agent manages scheduling process on behalf of its user by considering its own calendar and preferences. The location agent holds information on meeting rooms and actively takes part in the negotiation process.

Our system does not have a fixed central control. This means that there is not a specialized control agent and each agent is able to schedule a meeting via negotiation. Thus, a personal agent can be in the "organizer" role when its user requests a meeting and coordinates the meeting scheduling process, or it can be in the "invitee" role, as a participant of a meeting. Several meetings can be undergoing scheduling. Therefore, an agent can simultaneously be involved in scheduling any number of meetings, acting as an organizer for some and an invitee for others.

The proposed scheduler is implemented on JADE (Java Agent Development) [5] framework. JADE provides a distributed runtime environment in which agents can survive and communicate, also provides parallel, concurrent and multiple working agents through behaviour activities. It also presents a graphical user interface and debugging tools to be used for monitoring, logging and managing agents. Furthermore, designing and developing an agent system is easy due to its wide library. Most important, JADE is compliant with the Foundation for Intelligent Physical Agents (FIPA) specification, hence it can interoperate with other agents that conform to the same standards at the development phases [5, 6, 7]. FIPA, standards organization for agents and multi-agent systems was officially accepted by the IEEE as its 11th standards committee on 8 June 2005 [8].

In Section 2, we describe multi agent system architecture and the details of the agent implementation. In Section 3, we present the protocol that is used in the scheduling process. Section 4 discusses experiments on scheduling time and efficiency. Conclusion about the work takes part in Section 5.

2 Multi Agent System Architecture

As meeting scheduling is naturally a distributed processing, agents are distributed in our solution. As stated earlier, personal agents, one associated with each user, and a location agent are involved in the scheduling process. Each personal agent has access to its user's calendar and preference information, which is kept at different sites due to security issues, in accordance with the distributed nature of the problem. The location agent holds information on meeting rooms and generates responses to requests from personal agents. The location agent has its distinct calendar, holding data for already scheduled meetings and meeting proposals that are still in the negotiation phase. Furthermore, it also owns data storage for physical features of meeting rooms, such as capacity and equipments present. There is only one location agent in the system. Agents negotiate by having one agent propose a meeting, which the other agents accept or reject, based on whether or not it fits their own schedules. A meeting is represented with a start, end date, a start time, duration, and sometimes a type, meeting room and it is scheduled when all agents reach an agreement on values for these attributes.

We have classified invitees into two groups, important invitees and regular invitees, to meet real world requirements. Important invitees have to attend a meeting; therefore a meeting can only be scheduled only if all of the important invitees agree on a time slot. Attendance of regular invitees is not a necessity. The system has pre-defined meeting types held in the location agent's data store. Agents may negotiate for a particular type of meeting.

The negotiation process for scheduling has two stages. In the first stage, the organizer agent communicates with the location agent to send an appointment proposal. The location agent looks up its calendar to find out if the time interval is convenient. If it is, it accepts the proposal and blocks the proposed time slot in its calendar. If not, it tries to find a counter proposal time slot by using the features of the meeting and its calendar. It sends the organizer agent a message that contains the counter proposal if it can find a free time slot. In the second stage, the organizer agent announces the appointment proposal to all types of invitees and waits for their responses, which can be either an accept, a reject or a counter proposal. Next, the organizer agent evaluates the responses and decides on to start a new iteration with a new proposal or to cancel the meeting, and sends information messages to the invitee agents involved about the success or termination of the negotiation so that they can take appropriate action to update their calendars accordingly. Both types of agents can carry out actions for more than one negotiation concurrently.

Our work also supports for cancellation of already scheduled meetings. A user can view his/her calendar and cancel an appointment. However, only the organizer or an important invitee is allowed to carry out a cancellation operation.

As a supplement, a chat module, implemented in an agent oriented way, permits users to communicate with instant messages. Additionally, a reminder can be set up to prompt meeting information to invitees and the organizer, at a pre-specified time before the meeting is due.

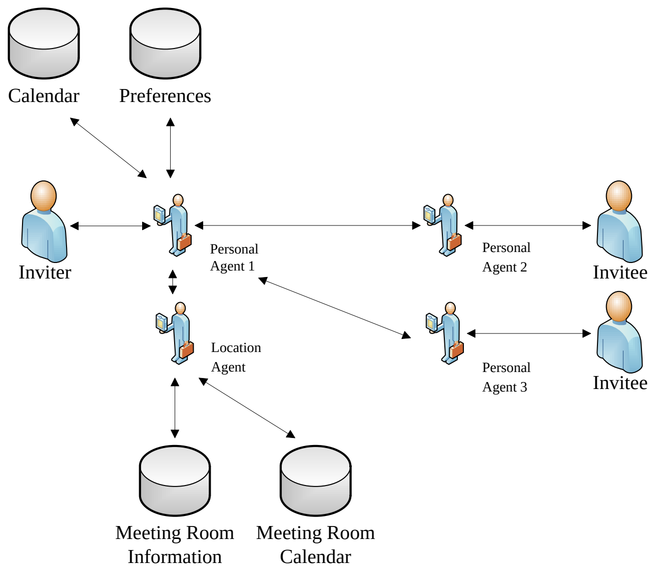

Fig. 1 Meeting Scheduling Model

The user provides the following information to the personal agent before negotiations start: the start date and the end date of the calendar block when the meeting is to take place, duration of the meeting, the start time, and an invitee list specifying the important and regular attendees. Optionally, information on meeting type, reminder time, a preferred location or the requested capacity and features of the location may also be provided. Inter agent communication paths of a sample scheduling process with an inviter and two invitees can be seen in Figure 1.

2.1 Personal Agent

A personal agent may be acting as an organizer, starting and coordinating a new scheduling negotiation process, or acting as an invitee, responding to meeting requests from other personal agents. It may be playing both roles simultaneously as well.

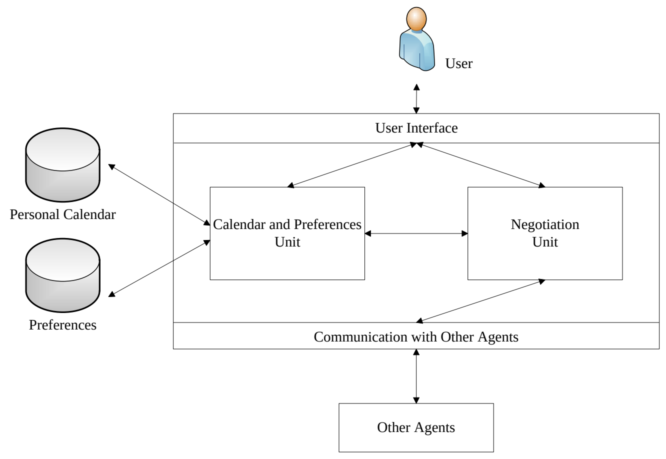

Fig. 2 Personal Agent

The inner structure of a personal agent is depicted in Figure 2. The user interface is the human interface to the system where data and requests on meetings and also on preference and calendar information are received. Through this interface, a user can initiate a new negotiation, view/cancel planned meetings and change his/her preferences.

The calendar database holds personal calendar information with time intervals allocated for meetings that are either finalized or in the negotiation phase. The preference database holds user's meeting preferences on calendar days or time intervals. A preference may be of type “strict”, specifying periods when the user wants no meetings to be scheduled. For instance, “Monday between 12:00 and 14:00 lunchtime” is a strict preference. A preference may also bring restrictions on meeting types, specifying periods when the user would accept only meetings of a certain type. An entry such as “Tuesday between 14:00 and 18:00 - coordination meetings” will only allow meeting requests with type equal to “coordination” get a positive response.

The Calendar and Preferences Unit produces appropriate results to query requests it receives from the Negotiation Unit and from the user. It also receives data to update the databases.

The Negotiation Unit receives user requests over the user interface and coordinates the scheduling process with close cooperation with the Calendar and Preferences Unit and other agents in the system. If the agent is acting as an invitee, the unit generates messages of acceptance, rejection or counter proposal according to results received from queries on the calendar and preferences.

2.2 Location Agent

In the system, there is a single location agent whose task is to respond to requests for meeting rooms. The inner structure of the location agent is depicted in Figure 3. The Calendar database keeps information on all scheduled meetings on all meeting rooms and reservations for meeting negotiations that are in progress. Meeting Room Information database holds the names of meeting locations along with their capacity and features, such as equipment present (white board, data show, computer, etc.). The User Interface allows the “administrator” to add new locations and/or to modify the features of the locations present in the database.

The Calendar and Meeting Room Unit receives queries and update requests to query and update the relevant databases. The Negotiation Unit receives messages from personal agents through the course of a negotiation. Requests come from personal agents that are acting in the organizer role, either questioning the availability of a certain location or requesting a location with certain features. If a specific meeting location is demanded, the unit checks the Meeting Room Calendar to see if it is convenient for the proposed time interval. If not tries to find a new time slot between the start and end date, for that location and generates a counter proposal.

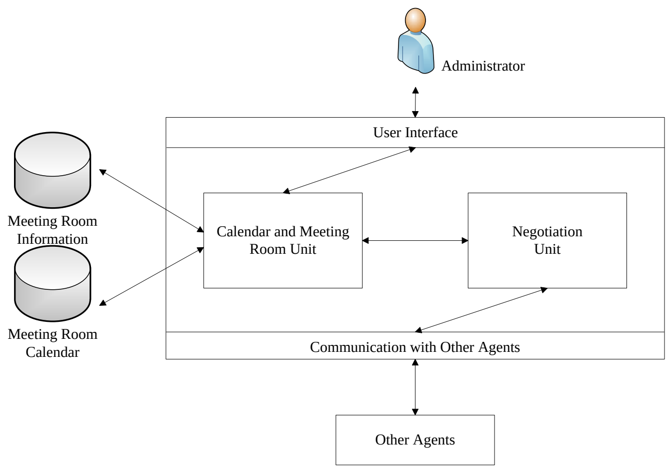

Fig. 3 Location Agent

If a particular meeting location is not indicated but certain features are specified, the Negotiation Unit interacts with the Calendar and Meeting Room Unit to find a suitable place. The result is sent as a proposal back to the inviter agent.

3 Protocol

Before presenting the meeting scheduling algorithms, we want to focus on certain decision criteria that influence the efficiency and speed of the scheduling process. The first criterion is related to the amount of information exchanged between the agents during the negotiation. If the agents exchange all of their calendar and preference information, the scheduling problem could be solved in a single iteration; however the user privacy principle would be violated. If the agents simply reply to meeting proposal with yes/no answers, there will not be privacy violation; however, the number of iterations that is necessary to complete the negotiation will rapidly increase. The privacy and efficiency issues must be balanced [9]. In our work, the agents are active so that their answers are not just yes or no. If a received proposal is not suitable, the agent tries to find an alternative date that fits its calendar and sends a counter proposal instead of a plain reject message. This approach furnishes the organizer agent with valuable information and reduces the number of future iterations needed for the meeting be scheduled.

A second criterion that has to be considered is the search technique over the calendars. Agents search their calendars for alternative dates when received proposals are not suitable. Linear early, linear least dense and hierarchical are search techniques that can be used [10]. In this study, we prefer linear early search technique. When a proposal is not convenient, the agent looks for the earliest possible interval in the calendar, skipping over any intervals overlapping with already scheduled meetings, and produces a counter proposal with the earliest free interval on the calendar that is long enough to match the request.

As agents can simultaneously be involved in multiple conversations with other agents, the same time slot may be the target of different proposals, possibly resulting in the arrangement of meeting that overlap in time. To cope with this situation, the negotiation units take the necessary actions that result in the blocking of the time intervals (reservations) for which the negotiation process has not yet terminated. If a message confirming that the meeting has been scheduled with success arrives, the interval related to that meeting is blocked permanently. If, on the contrary, the message withdraws the proposal, the block on the interval is removed so that it can be considered in forthcoming negotiations. Again, when a proposed time interval for a particular meeting is changed due to the processing of a counter proposal, all related past reservations are removed from the calendar.

The scheduling protocol has two stages; in the first stage, the organizer agent tries to come to an agreement with the location agent. In the second stage, the organizer announces the proposal and starts to negotiate with the invitees. The detailed flow of the protocol after the agent in the organizer role receives the appointment data from its user is as follows:

1. The proposed meeting time is checked for suitability on the user’s calendar and preferences databases. If not suitable, the agent proceeds to determine the earliest suitable time between start and end dates. If the search does not terminate successfully, the scheduling process is cancelled as no suitable time has been found for the meeting. Otherwise, the agent rebuilds the proposal with this new information, blocks the interval on the calendar, and sends the proposal to the location agent in order to reach to an agreement on the location of the meeting.
2. When location agent gets the proposal, it checks to see if a particular meeting room is specified.
    a. If not, it extracts the data on equipments and capacity from the message and then looks for a meeting room that matches and is free at the specified time. If it can provide a location at the specified time, it sends an accept message, or if it can provide a location at another time in the calendar, it sends a counter proposal. If both of these are not possible, a reject message is sent.
    b. If a meeting room is specified in the proposal, the location agent looks at the calendar to see if that place is free for the given interval. If it is, it sends an accept reply to organizer. Otherwise, it tries to find a new time interval when that meeting place is free. If successful, it responds with a counter proposal, including the new meeting time; otherwise, a reject message is sent.
3. Organizer agent receives the location agent’s reply.
    a. If it is a reject message, the scheduling process is cancelled as no room could be allocated for the meeting. Otherwise, the agent announces the proposal to the invitees.
    b. If the received message is a counter proposal, it refers to its calendar and preferences for any conflicts. If suitable, the agent modifies its proposal accordingly and announces the proposal to the invitees. If not, the agent returns to Step 1 to produce a new proposal.
4. An invitee agent which receives a proposal refers to its calendar and preferences to find out if the time interval is free. If it is free, the interval is reserved and an accept proposal message is generated.

Otherwise, it tries to find a new time interval. If successful, it responds with a counter proposal, including the new meeting time; otherwise, a reject message is sent.

5. Organizer agent receives reply messages from all of the invitees and evaluates them.
    a. If all of the important invitees have accepted the proposal, organizer sends confirmation messages to those invitees who have accepted the proposal and cancellation messages to regular invitees who have rejected it. A confirmation message finalizes a negotiation and causes a reserved calendar interval to be blocked. A cancellation message, on the other hand, ends the negotiation and frees reserved intervals.
    b. If one or more reject messages are received from important invitees, the organizer sends cancellation messages to the location agent and all the invitees.
    c. If no reject messages come from important invitees but one or more counter proposals are received, the organizer determines the counter proposal that has the time interval farthest in time. If that interval is suitable, the agent proceeds from Step 1. If the interval is not free but an new proposal can be generated, the agent again continues with Step 1.On the other hand, if the search for a new proposal fails, the organizer sends cancellation messages to the location agent and the invitees.

4 Experiments

We have carried out two experiments to show how scheduling time and efficiency of the negotiation protocol vary under different calendar densities and different number of meetings. The experimental execution environment consists of 8 personal agents and a location agent. The calendars of the users are fixed to 5 working days, with 45 slots (9 slots for each day). No preference data is used in the experiments and it is assumed that meeting locations have adequate equipment and capacity. Meetings for each experiment are randomly generated. The scheduling processes for all meeting requests start and execute concurrently.

4.1 Experiment 1

The goal of this experiment is to measure the effect of calendar density on the number of successfully scheduled meetings. The experimental procedure proceeds to schedule new meetings on calendars of different density. The precondition calendar density is obtained as follows: for instance to produce a calendar density of 10, we create 10 meeting requests randomly, one with at least 7 invitees, one with 6 invitees, and others with 5, 4, 3 and 2 invitees. Next, those meetings are scheduled to produce a calendar density of 10. In the experiment, calendar densities vary from 10 to 80. As the density increases, the number of meetings with different numbers of invitees increases linearly.

The experiment involves the scheduling of 10 more new meetings on calendars of different densities. Those 10 meetings are also created randomly.

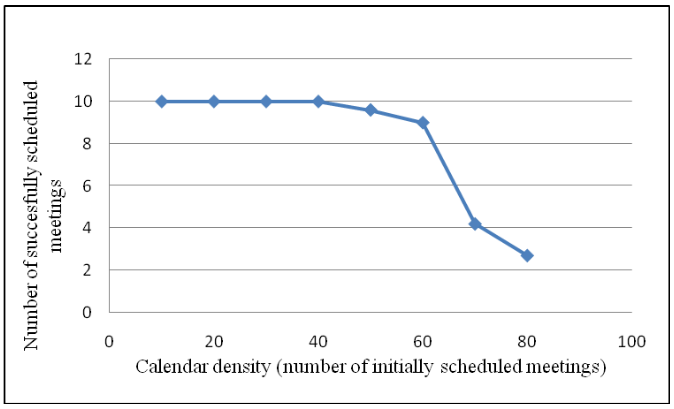

Fig. 4 Scheduling on different calendar densities

Figure 4 shows the number of successfully scheduled meeting for calendars of different densities. We see that up to a calendar density of 60, the ratio of initiated to successfully scheduled meetings is nearly 100 %, but for higher calendar densities, the ratio decreases dramatically. This is because the numbers of free time slots the agents have in common decreases and also concurrent negotiations block several time intervals at the same time, leaving no alternatives for counter proposals.

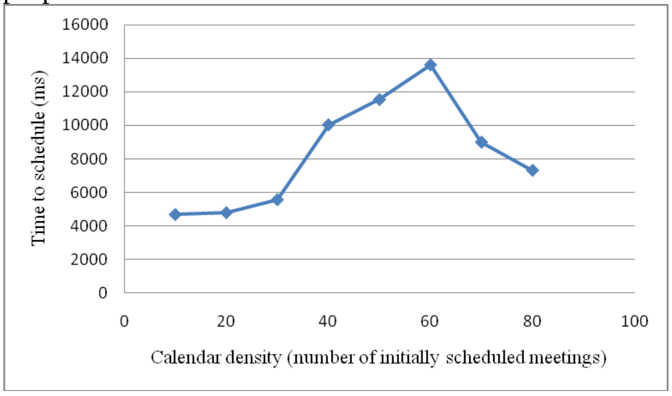

Fig. 5 Relation between scheduling time and density

In Figure 5, we can see the total time required to complete negotiations on different calendar densities. Scheduling time increases almost linearly until a density of 60 because the denser the calendar, the greater is the communication overhead of the negotiation. However, after the boundary of density of 60, there is steep fall in scheduling time, mainly resulting from negotiations that fail to schedule a meeting. As agents cannot find free slots, they send rejection messages that end the negotiation, hence minimizing negotiation time.

4.2 Experiment 2

The goal of this experiment to observe the performance of the system when scheduling load increases. The density of the calendar is fixed to 30 meetings and the number of meeting to be scheduled is varied from 10 to 80. We measure the number of successfully scheduled meetings and the time consumed by negotiations.

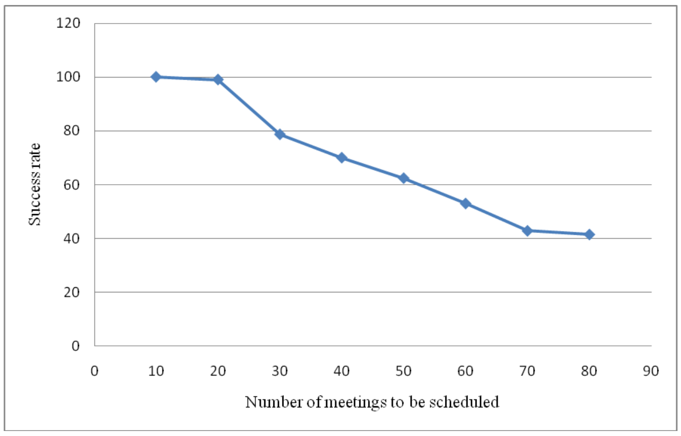

Fig. 6 Success rate for different numbers of meetings

Figure 6 shows the success rate as the number of meetings to be scheduled increases. We see 100% success up to 20 meetings, as they all are successfully scheduled. After this point, the success rate linearly decreases, because the number of common free time slots falls and, also, concurrent negotiations block a large number of time intervals at the same time. If we look at the number of successful negotiations for the worst cases, we see the number is nearly 30. That is because the calendars become nearly full after 30 meetings more and do not have room for additional meetings.

Figure 7 shows the measured time needed to schedule different numbers of meetings, again under a calendar density of 30. In Figure 6, we observed that when the try to schedule 30 to 80 meetings, the number of successfully scheduled meetings is always about 30. Even though this is still the case in this experiment, we observe that the scheduling time increases in a linear way. This is because of the time taken by negotiations that have completed unsuccessfully.

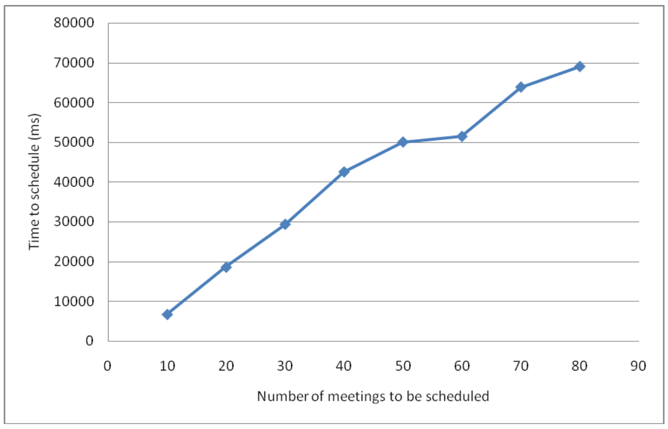

Fig. 7 Scheduling time for different numbers of meetings

5 Conclusion

This paper describes the design and implementation of a distributed multi agent meeting scheduling system. The model we propose includes personal agents that may be both in organizer and invitee roles and a location agent that handles requests for meeting rooms. Invitees are characterized as either important or regular and influence negotiation decisions accordingly. We describe two staged protocol we have implemented to handle the scheduling problem. The protocol considers user preferences and allows invitees to generate counter proposals, from which the organizer can make inferences and not send redundant proposals. We give the results of experiments carried out to measure the performance and efficiency of the protocol.

References:

[1] Sen, S., and Durfee, E. H., A formal study of distributed meeting scheduling. *Group Decision and Negotiation*,1998, pp. 265–289.

[2] Franklin S.,Graesser A., Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents, *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, 1996, pp. 21- 35

[3] Scott A., Jenkin K., Senjen R., Design of an Agent-Based, Multi-user Scheduling Implementation, *Selected Papers from the 4th Australian Workshop on Distributed Artificial Intelligence, Multi-Agent Systems:Theories,Lang.,&Applic.*,1998, pp. 152-165

[4] Crawford E., Veloso M., Opportunities for Learning in Multi-Agent Meeting Scheduling, *In Proceedings of the AAAI*, 2004

[5] Bellifemine F., Caire G., Poggi A., Rimassa G., JADE A White Paper, *Telecom Italia EXP magazine Vol 3,No 3 September*, 2003

[6] Caire G.,JADE, Programming for Beginners,*Telecom Italia*, 2003

[7] Bellifemine F.,Caire G.,Trucco T.,Rimassa G., JADE Programmer's Guide, *Telecom Italia, 2005*

[8] FIPA specifications, http://www.fipa.org

[9] Freuder E.C., Minca M., Wallace R.J., Privacy/efficiency tradeoffs in distributed meeting scheduling by constraint-based agents, *Proc. IJCAI–01 Workshop on Distributed Constraint Reasoning*, 2001, pp. 63–71.

[10] Sandip Sen and Edmund H. Durfee, Unsupervised Surrogate Agents and Search Bias Change in Flexible Distributed Scheduling. Proceeding, *1st. International Conference on Multi-Agent Systems p.336-343 San Franscisco*, 1995, pp. 336-343