Title
Distributed Meeting Scheduling

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 16(0)

Authors
Liu, JyiShane
Sycara, Katia P.

Publication Date
1994

Peer reviewed

Distributed Meeting Scheduling

JyiShane Liu and **Katia P. Sycara**
The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
jsl@cs.cmu.edu and katia@cs.cmu.edu

Abstract

Meeting scheduling takes place when a group of people intend to meet with each other. Since each person has individual availability constraints and preferences, meeting scheduling is naturally distributed and there is a need to schedule the meeting in such a way as to consider the preferences of the set of meeting participants. In addition, individual meeting constraints and preferences may change both as a result of an agent's situation or as a result of other agents' scheduling decisions. Therefore, there is a need for distributed reactive schedule revision in response to changing requirements and constraints. We present an approach to distributed meeting scheduling based on modeling and communication of constraints and preferences among the agents. When a feasible global schedule cannot be found, agents enter a negotiation and relax their constraints. The approach enables the agents to find and reach agreement on the schedule with the highest joint utility and to reactively revise the schedule in response to new information.

Introduction

Advances in computer and networking technology have fostered the development of the needed infrastructure to provide automated support for group decision making in organizations. AI techniques can introduce "intelligent agents" into organizational computing systems that act in the interests of their human clients to perform routine organizational tasks, such as screening, directing or responding to information (Lee & Malone, 1988; Sycara & Roboam, 1991). Scheduling meetings is one of the most frequently performed tasks in an organization. It is often tedious, iterative and time consuming for people. In addition, because of communication delays (either in electronic mail or phone), the task can be frustrating and produce unsatisfactory solutions. Therefore, automating meeting scheduling can lead to more satisfying and efficient solutions and to changes of how information is exchanged within organizations (Feldman, 1987). Past research efforts (e.g., (Greif, 1982; Lee & Malone, 1988)) have met with limited success. DAI work in meeting scheduling (Sen & Durfee, 1993) focused on various search biases to produce alternative density profiles in agents' calendars. In their model, a fixed meeting's host communicates with all other agents and coordinates the search for a feasible schedule without taking into consideration individual preferences or dynamic constraint changes. Commercially available products (e.g. Schedule Plus) are simple checkers of availability of time intervals and are incapable of taking into account the preferences and priorities of people, or reactively revising a schedule in response to new information.

Meeting scheduling takes place when a group of people intend to meet. Since each person has individual availability constraints and preferences, meeting scheduling is naturally distributed and must consider the preferences of the set of attendees. Distributed scheduling is a process carried out by a group of agents each of which has (a) limited knowledge of the constraints of other agents and (b) limited knowledge of the preference of other agents. Global system solutions are arrived at by interleaving local computations and information exchange among the agents. There is no single agent with a global system view through the entire scheduling process. In such an environment, schedules are constructed by a process of reaching consensus. Agents make local decisions about assignments of particular time intervals to particular activities and a complete schedule is formed by incremental agreement. The system goal is to find schedules that are not simply feasible but attempt to create high quality schedule according to the group's collective preferences. Another, equally desirable requirement is schedule revision. In most realistic environments unanticipated events occur (e.g. a meeting might last longer than expected, or a meeting participant might not show up) that necessitate schedule revision. Our approach endeavors to address concerns of schedule quality and efficiency of schedule generation, and schedule revision.

Our work views meeting scheduling as a distributed task where a separate calendar management software agent that knows its owner's preference is associated with each person and acts on behalf of its user. Giving each person his or her own calendar management agent enhances privacy and permits personal tailoring of preference parameters for scheduling meetings (Dent *et al.*, 1992). Individual meeting constraints and preferences may change both as a result of an agent's situation and also as a result of other agents' scheduling decisions (Sycara *et al.*, 1991). Optimization criteria are expressed in terms of preferences/utilities for particular scheduling decisions (e.g., prefer earlier start times for activities). A feasible schedule is one that satisfies all given constraints. The agents exchange scheduling constraints and preferences in a joint search for the best schedule.

The rest of the paper is organized as follows: section 2 presents the distributed meeting scheduling model, coordination mechanism as well as communication protocol for multi-agent meeting scheduling; section 3 describes an example of meeting scheduling scenario; section 4 presents discussion; and section 5 presents conclusions.

The Distributed Meeting Scheduling Model

Each human meeting attendee is supported by a software agent that manages its owner's calendar. A meeting has a date and location. A meeting schedule is feasible, if it is agreed upon by all agents representing the attendees.[1] The group common goal is to generate a feasible schedule that is the most preferred by the participants. Our model also allows for representing and enforcing time dependencies between meetings. For example, the user can specify that meeting A should be scheduled after meeting B. We distinguish between two types of constraints: *task precedence* constraints expressing sequencing relations between meetings and *resource capacity* constraints that express time availability of meeting attendees. Task precedence constraints are not relaxable. The resource capacity constraints of a meeting attendee restricts the dates on which s/he is available. In our model, resource capacity constraints can be relaxed as a result of negotiation.

With each task, we associate utility functions that map each possible date onto a preference. These utilities reflect an attendee's meeting date preferences and can change dynamically depending on other scheduled meetings or other external circumstances. In the cooperative setting assumed in this paper, the sum of these preferences over all the agents in the system and over all the tasks to be scheduled by each of these agents defines the joint group utility.[2] Agents will prefer a solution that gives higher joint utility to a solution of lower joint utility.

The group of agents work together to generate a schedule that is feasible, agreeable to all and has the highest possible joint utility. Each agent searches its individual search space to find a solution that reflects its owner's preferences. The agents communicate with each other to direct each other's search. Search focusing heuristics inspired by experimental studies of how humans handle scheduling (Kelley & Chapanis, 1982), such as constraint tightness, are utilized. The group has no quarantee that a schedule that attains the highest value of joint utility will be found but the model gives each one of the participants the flexibility to object if a proposed schedule has low utility for the attendee. The model also gives the paricipants a metric (i.e., the group utility) by which to gauge the global group preference.

Coordination Mechanism

In a multi-agent distributed scheduling setting, coordination between agents is essential for effectively generating a feasible and high utility schedule. In our approach, global search is directed by a coordination mechanism which dynamically passes search control to different agents according to policies that are mutually accepted and adhered to by the agents. [3]

The coordination policies are:

1. In each round of constructing and modifying a solution, a task coordinator is selected for each task who proposes an assignment for the task and receives replies.
2. A task coordinator broadcasts its coordinating role to all other concerned agents.
3. A task coordinator sends other task coordinators proposals for solving the task.
4. Initially, agents with the tightest resource capacity constraints for each task are the task coordinators.
5. The role of next task coordinator for each task is assigned by the current task coordinator to the agent who opposes the current proposal. If there are more than one opposing agents, the agent with the tightest constraints is selected.
6. If a new task coordinator can not find a feasible solution, it should relax its constraints until it can propose a feasible solution.

Communication Protocol

The agents communicate asynchronously via message passing. In many problems, a series of meetings must be scheduled. The process of reaching a consensus on a schedule for a sequence of tasks (meetings) is realized by communication and negotiation. The negotiation proceeds by relaxation of local agent constraints (Sycara, 1990). The multi-agent meeting scheduling communication and negotiation protocol is as follows:

1. The agents exchange their resource capacity constraints, i.e. the time intervals on which they are available.
2. Agents with the tightest constraints for each task (i.e. the agents with the fewest available intervals) become the task coordinators and propose a schedule for the task by sending a message to the other related agents.
3. Depending on its local constraints, an agent receiving a proposal either accepts or rejects it by sending a message back to the proposing agent. If an agent accepts a proposal, a utility representing the agent's preference measure on the proposal is associated with the replying message.
4. The task coordinators wait to receive all replying messages.
   - If a proposal is accepted by all, a task coordinator sums up the utilities included in the replies as well as its own utility to get the global preference measure for the proposal, and notifies the related agents about the common approval of the proposal and its global utility.
   - If a proposal is rejected by at least one agent, a task coordinator selects the next task coordinator according to the defined policy and delegates the role by sending a message to that agent.
5. New task coordinators relax their local constraints, when necessary, provide new proposals and repeat the process of coordinating the group decision-making.
6. A feasible schedule is generated when all tasks have been scheduled and approved by all relevant agents. Each feasible schedule that includes all the tasks has a global preference measure which is the sum of the preference measures of the schedule for each task.
7. Due to local reasons, an agent can contest the current feasible schedule and ask for further negotiation. If all agents agree to further negotiate, agents who contested the current feasible schedule become the task coordinators.

[1]Naturally, the agents interact with their human owners to get needed approvals. We do not address HCI concerns here.

[2]For simplicity of exposition, we say "an agent's preference" as a short hand for "the preference of an agent's human owner"

[3]Cammarata & Steeb's work on distributed coordination in the air traffic domain (Cammarata & Steeb, 1983) has experimented with different types of criteria for task coordinator assignment. In that work, however, once assigned, the coordinator role was statically associated with an agent.

8. When an agent has local constraint changes, it assumes the role of task coordinator to further improve the schedule when additional dates are available or to find another feasible schedule when the scheduled dates are now unavailable.
9. The current active schedule is a feasible schedule with the highest global preference.

We have identified 14 message types that are sufficient to cover all possible agent exchanges under the protocol. They are:

1. **Availability_Constraint**($dates$) - notifying receiving agent about the available dates of the sending agent.
2. **Task_Coordinator**($task_i$) - notifying receiving agent that the sending agent is the task coordinator for ($task_i$).
3. **Proposal**($meeting_i$, $date_j$) - notifying receiving agent about the current proposed solution, $date_j$, for $meeting_i$.
4. **Accept**($meeting_i$, $date_j$, $preference_k$) notifying receiving agent that the current proposed solution, $date_j$, for $meeting_i$, is accepted by the sending agent with $preference_k$.
5. **Reject**($meeting_i$, $date_j$) - notifying receiving agent that the current proposed solution, $date_j$, for $meeting_i$, is rejected by the sending agent.
6. **Delegate**($meeting_i$) - notifying receiving agent that it has been delegated as the task coordinator for $meeting_i$.
7. **Approved**($meeting_i$, $date_j$, $preference_k$) - notifying receiving agent that the current proposed solution, $date_j$, for $meeting_i$, has been approved by all related agents with global preference, $preference_k$.
8. **Announce**($meeting_i$, $location_j$) - announcing to receiving agent that $meeting_i$ will be held at $location_j$ based on common agreement.
9. **Abandon**($meeting_i$) - notifying receiving agent that the effort to find an optimal schedule for $meeting_i$ is abandoned.
10. **Negotiate?**($schedule_i$) - contesting $schedule_i$ and asking receiving agent whether it agrees to further negotiate for possible schedules.
11. **Negotiation_OK**($schedule_i$) notifying receiving agent that the sending agent agrees to negotiate on $schedule_i$.
12. **Negotiation_Denied**($schedule_i$) - notifying receiving agent that the sending agent does not want to negotiate on $schedule_i$.
13. **Preference_Changes**($meeting_i$, $date_j$, $preference_k$) - notifying receiving agent that the preference measure for the accepted proposal is changed to $preference_k$.
14. **Constraint_Changes**(*increase/decrease, dates*) - notifying receiving agent that constraints of the sending agent are changed with an increase/decrease of availability on some particular dates.

These 14 message types relate to different aspects of the multi-agent meeting scheduling communication and negotiation protocol. **Availability_Constraint** is used by agents to communicate local constraints. **Task_Coordinator** and **Delegate** facilitate dynamic search control. Agents use **Proposal**, **Accept**, and **Reject** to negotiate a feasible solution, and use **Abandon** to terminate a negotiation process. **Approved** and **Announce** allow agents to communicate current feasible solution and auxiliary information regarding current feasible solution, respectively. **Negotiation?**, **Negotiation_OK**, and **Negotiation_Denied** are used by agents to reach a consensus before entering a new negotiation stage that would potentially change the current feasible solution. Finally, **Preference_Changes** and **Constraint_Changes** allow agents to express changing preferences and constraints, and therefore, reactively revise the solution in response to new information.

Characteristics of the Approach

Our approach has several characteristics that make it attractive. First, global planning is distributed to each agent, while the control is dynamically passed to different agents according to the coordination policies and the changing interaction context. Having a task coordinator has the advantage of minimizing communication overhead compared with completely distributed coordination (e.g., Partial Global Planning (Durfee, 1988)), at the same time, allowing flexibility (since the task coordinator role is dynamically passed among different agents) as compared with centralized coordination. Second, only local constraints are exchanged and no agent has information on local preferences of other agents. Moreover, constraints that are not directly relevant to a task are not exchanged. For example, agents that participate in the same meeting exchange their available dates for that meeting only. The constraint communication mechanism aims at minimizing communication and maximizing agent privacy. Third, since scheduling is an NP-complete problem (Garey &Johnson, 1979), in general, it takes exponential time to find an optimal solution. The assumption and coordination mechanism of the model enable the attendees to reach meeting agreements with acceptably high individual utility as well as high joint utility. Coordination policies define the focus of the search with the notion of tightest constraints directing the search a for feasible solution and the discontented agent(s) directing the search for solutions of higher utility. These are strategies often observed in human negotiation (Kelley & Chapanis, 1982). Fourth, the approach can reactively adapt the schedule in response to opportunities and harmful events. When opportunities occur, e.g., additional dates become available, agents seek to improve the quality of the solution. When harmful events occur, e.g., scheduled dates become unavailable, agents seek other feasible schedules through negotiation.

An Example - The Secretaries' Nightmare

We illustrate the approach through an example that is part of daily life in organizations.[4] Axel, Brigitt, Carl, and Dirk wish to plan two all-day meetings. The meetings must be scheduled on weekdays during April 1994. Axel, Brigitt, and Dirk should attend the first meeting. Axel, Brigitt, and Carl should attend the second meeting. If possible, scheduling two days in a row is preferable. If the meetings are not scheduled back-to-back, Carl and Dirk prefer to have a meeting before the second meeting so that Dirk can brief Carl about the results of the first meeting. The faster the meeting can take place the better. The meeting can be in either Austin or Los Angeles, but it is preferable for the second meeting to be in Los Angeles. If the meetings are back-to-back in the same city, Los Angeles is preferable. Axel is available in April the week of the 4th, the 18th and 19th, and the 25th and 26th. Brigitt is available

[4]This example was suggested by Charles Petrie.

the 7th, 8th, 19th, and the week of the 25th. Carl is available on the 7th, 19th, and 26th. Dirk is available on the 7th, 8th, 18th, and 25th.

This scenario has three tasks (meetings): two mandatory, Meeting-1 and Meeting-2 and one optional Meeting-3 between Carl and Dirk if Meeting-1 and Meeting-2 are not scheduled back to back. In our model, we assume that agents know which meeting they must attend and the sequencing meeting constraints, i.e. Meeting-1 must be scheduled before Meeting-2. In addition, the group shares the preference for meetings as early as possible as well as the preference for meeting locations. At first, all agents participating in the same meeting exchange their available dates.[5] For example, Dirk sends **Availability_Constraint**(7,8,18,25) to Axel, Brigitt, and Carl, Carl sends **Availability_Constraint**(7,19,26) to Axel, Brigitt, and Dirk, and so on. After receiving the exchanged constraints, each agent knows who is the agent with the tightest constraints, i.e. the fewest available dates, among the attendees of a particular meeting. In the scenario, Dirk is the agent with the tightest constraints among the attendees of Meeting-1, and Carl is the one for Meeting-2. Therefore, Dirk sends **Task_Coordinator***(Meeting-1)* to Axel and Brigitt, announcing his coordinator role for Meeting-1. Carl does the same to Axel and Brigitt with the message **Task_Coordinator***(Meeting-2)*.

Meeting-1	
Attendees	Available Dates
Axel	4 5 6 7 8 18 19 25 26
Brigitt	7 8 19 25 26 27 28 29
Dirk	7 8 18 25

Meeting-2	
Attendees	Available Dates
Axel	4 5 6 7 8 18 19 25 26
Brigitt	7 8 19 25 26 27 28 29
Carl	7 19 26

Figure 1: Availability of meeting attendees

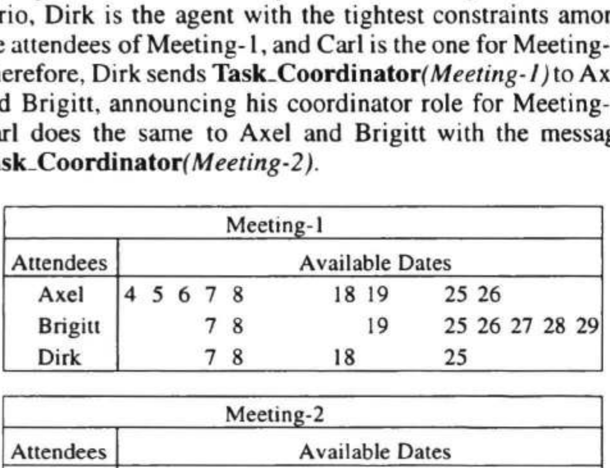

Figure 2: Utility functions of preferences

Carl, Dirk, Axel and Brigitt prefer that the meetings be held as early as possible. We assumed that each agent has a preference measure for each available date on an arbitrary scale of 0 to 10 (0 the lowest and 10 the highest). The group preference of having a meeting as early as possible is expressed for each agent by giving a 10 to the agent's first available date and decreasing the value incrementally for his/her subsequent available dates. For example, Axel would have a preference measure of 10 for the 4th, 9 for 5th, ..., 2 for the 26th, Carl would have a preference measure of 10 for the 7th, 9 for the 19th, and 8 for the 26th. For simplicity, we have given all agents the same highest utility for the first available date and decreased the utility by the same amount for each subsequent date.

Dirk, knowing the constraints of Axel and Brigitt (and the group preference for early meetings), decides that the best solution for Meeting-1 is on the 7th. Dirk sends **Proposal***(Meeting-1, 7)* to Axel, Brigitt, as well as Carl (since Dirk and Carl must also coordinate to decide on optional Meeting-3). In the mean time, Carl, knowing that Meeting-2 should be held after Meeting-1, awaits the decision on Meeting-1. After receiving Dirk's proposal on Meeting-1, Carl decides that the best date for Meeting-2 is on the 19th. He, then, sends **Proposal***(Meeting-2, 19)* to Axel and Brigitt. Axel, while receiving proposal messages from Dirk and Carl, checks his local calendar. Since the proposed dates are still available, Axel sends **Accept***(Meeting-1, 7, 7)* to Dirk and **Accept***(Meeting-2, 19, 4)* to Carl. 7 and 4 are Axel's preference measures for proposed meeting dates for Meeting-1 and Meeting-2, respectively. Brigitt also sends **Accept** messages to Dirk and Carl with preference measures of 10 and 8, respectively.

Receiving positive replies from both Axel and Brigitt, Dirk sends **Approved***(Meeting-1, 7, 27)* to Axel, Brigitt, and Carl, while Carl sends **Approved***(Meeting-2, 19, 21)* to Axel, Brigitt, and Dirk. In the first approval message, 27 is the total preference measures of Axel, Brigitt, and Dirk for having Meeting-1 on the 7th (Dirk's preference for the 7th is 10). Similarly, 21, in the second approval message, is the total preference measures of Axel, Brigitt, and Carl for having Meeting-2 on the 19th. Having received approval message from each other, both task coordinators, Dirk and Carl, know that the meetings will not be held back-to-back and respectively send out **Announce***(Meeting-1, Austin)* and **Announce***(Meeting-2, Los Angeles)* to others. Now all agents understand that they have a meeting schedule in which Meeting-1 is to be held at the 7th in Austin and Meeting-2 is to be held at the 19th in Los Angeles. This global schedule, labeled Plan-1, has a global preference measure of 48 (the sum of the preferences of the two meeting schedules).

Under Plan-1, Dirk and Carl recognize that they would need an optional meeting, Meeting-3, between them. Since Carl has tighter constraints than Dirk, Carl assumes the role of coordinator for Meeting-3 and sends **Task_Coordinator***(Meeting-3)* to Dirk. Carl finds out that he does not have any date available between the 7th and the 19th, and he would not relax his constraints for the optional meeting. He sends **Abandon***(Meeting-3)* to Dirk. Then, they both contest the current feasible schedule, Plan-1, by lowering their preference measures such that the total preference measure for Meeting-1 becomes 22 and that for Meeting-2 becomes 16. They individ-

[5] For clarity, we refer to a human's software agent by its owner's name.

ually send **Preference_Changes**(*Meeting-1, 7, 22*) and **Preference_Changes**(*Meeting-2, 19, 16*) to others. Now Plan-1 has a global preference measure of 38. Both Dirk and Carl send **Negotiate?**(*Plan-1*) to Axel and Brigitt. Axel and Brigitt agree to negotiate by replying with **Negotiate_OK**(*Plan-1*).

Meeting-1			
Attendees	Possible Dates		
Axel	4 5 6 7 8	18 19	25 26
Brigitt	7 8	19	25 26 27 28 29
Dirk	5 6 7 8	18	25

Meeting-2			
Attendees	Possible Dates		
Axel	4 5 6 7 8	18 19	25 26
Brigitt	7 8	19	25 26 27 28 29
Carl	5 6 7	19	26

Figure 3: Updated availability of meeting attendees

Since Carl and Dirk contested the current meeting plan, they each assume the coordinator role for the meetings again and respectively send out **Task_Coordinator**(*Meeting-1*) and **Task_Coordinator**(*Meeting-2*) to others. Dirk proposes the 25th for Meeting-1 and notifies Axel, Brigitt, and Carl. Then, Carl proposes the 26th for Meeting-2 and notifies Axel, Brigitt, and Dirk. Axel and Brigitt accept both proposals with low preference measures. Dirk and Carl notify other agents about the approval of their proposals and then announce that the meeting location is Los Angeles since the proposed meetings are back-to-back. Now they have another plan, Plan-2. However, Plan-2 has a global preference measure of only 33, which is lower than that of Plan-1. All agents understand that Plan-1, with the highest preference measure, is the active current plan for their meetings. Carl and Dirk request further negotiation and Axel and Brigitt agree.

Again, Dirk and Carl assume the coordinator role for Meeting-1 and Meeting-2, respectively. Each of them decides to relax his constraints and makes the 5th and 6th available. They both notify others about their constraint changes, e.g., **Constraint_Changes**(*increase, 5,6*). Then, Dirk proposes the 5th for Meeting-1 and Carl proposes the 6th for Meeting-2. Axel accepts both proposals with high preference measures. However, Brigitt, for her own reasons, decides that she can not make both dates available, and rejects both proposals. Since Brigitt is the only one rejecting each proposal, Dirk and Carl both delegate the roles of task coordinator for Meeting-1 and Meeting-2 to Brigitt by sending her **Delegate**(*Meeting-1*), **Delegate**(*Meeting-2*).

Brigitt, after announcing her roles of task coordinator for both meetings, finds that she would need to relax her constraints in order to propose a feasible schedule. She makes the 18th available and notifies the other agents about the constraint changes. Then, she proposes the 18th for Meeting-1 and the 19th for Meeting-2. The proposals are accepted by all other agents. Brigitt, then, announces the current global schedule, Plan-3, in which Meeting-1 is to be held on the 18th in Los Angeles and Meeting-2 is to be held on the 19th in Los Angeles. Since Plan-3 has a global preference measure of 40, which is higher than that of Plan-1 (which was 38), it becomes the current active plan.

Meeting-1			
Attendees	Possible Dates		
Axel	4 5 6 7 8	18 19	25 26
Brigitt	7 8	18 19	25 26 27 28 29
Dirk	5 6 7 8	18	25

Meeting-2			
Attendees	Possible Dates		
Axel	4 5 6 7 8	18 19	25 26
Brigitt	7 8	18 19	25 26 27 28 29
Carl	5 6 7	19	26

Figure 4: Another updated availability of meeting attendees

Then, Carl's constraints change and the 8th becomes available. He notifies others about the constraint change and re-proposes Meeting-3 on the 8th to Dirk. The proposal is accepted by Dirk, and both Carl and Dirk change their preference measures of Plan-1. Now Plan-1 regains its original preference measure of 48 and becomes the current active plan. Carl observes that there is an opportunity to further improve the current schedule because of his new availability. Carl asks for negotiation on current plan and all other agents agree. At this point, Carl, still having the tightest constraint, announces his role of task coordinator for Meeting-2 and proposes the 8th for it. Both Axel and Brigitt accepts the proposal. But Axel gives a low preference because of other personal arrangements. The plan does not have higher preference than the current active plan, Plan-1. At this point, no agent has any other intention to change the plan. Therefore, Plan-1 becomes their final plan, in which Meeting-1 is to be held on the 7th in Austin, Meeting-2 is to be held on the 19th in Los Angeles, and an additional Meeting-3 between Dirk and Carl is scheduled on the 8th in Austin.

Meeting-1			
Attendees	Possible Dates		
Axel	4 5 6 7 8	18 19	25 26
Brigitt	7 8	18 19	25 26 27 28 29
Dirk	5 6 7 8	18	25

Meeting-2			
Attendees	Possible Dates		
Axel	4 5 6 7 8	18 19	25 26
Brigitt	7 8	18 19	25 26 27 28 29
Carl	5 6 7 8	19	26

Figure 5: Final updated availability of meeting attendees

Discussion

Daily life group decision making on scheduling meetings is typically full of negotiations and dynamic revisions, as described in the scenario. The running example shows how our approach can provide automated support for the often tedious, iterative, and time-consuming task. The approach assumes a cooperative setting, mutual acceptance and adherence to the coordination policies by the agents, and an agreement on meeting precedence constraints and location constraints. The agents communicate the subset of their constraints that they want to make public. Based on the exchange of these *public* constraints, they all know who the most constrained agent is. At the beginning of a negotiation process by common policy, the most constrained agent assumes the role of task coordinator. During the negotiation process, a new task coordinator is designated solely by the previous task coordinator. This coordination mechanism allows agents to effectively search for a feasible schedule if it is possible. The search for solutions of higher utility is directed by the discontented agent(s). We are aware of the theoretical modelling issue of utilities being non-commensurate across group members. However, in this work, the utilities are used for ranking alternative solutions rather than for exact modelling. We also make two assumptions on agents' negotiation process that are in accordance with the behavior of human negotiators (Raiffa, 1982). First, agents prefer to relax their local constraints rather than fail to find a feasible solution. Second, agents would accept a feasible solution with low individual utility if the request for further negotiation is denied by others. Therefore, the negotiation process will be terminated when no agents express discontent with the current solution or at least one agent denies further negotiation.

Conclusions

We have presented a model of distributed negotiation for scheduling meetings carried out by intelligent software agents on behalf of their users. We believe that such an approach is better than scheduling via electronic mail since users only need to give their constraints and preferences to the system without being bound to the tedious communication and negotiation processes. Users can give full or limited authority to their agents regarding constraint relaxation. In case of limited authority, users would need to interact with their agents when necessary. We have presented the communication protocol and types of messages exchanged. Schedule evaluation and decision to relax constraints are distributed; schedule proposal (determination of intersection of available dates) and computation of schedule utility (for each task) is done by a coordinator. The role of coordinator is not determined statically at system design time but is passed dynamically to different agents during problem solving, thus avoiding well known problems of reliability and congestion of centralized decision making. The approach is currently being implemented. We intend to experimentally analyze the effectiveness of the approach compared with others, such as scheduling meetings via electronic mail.

Acknowledgements

This research is partially supported by the Defense Advance Research Projects Agency under contract #F30602-88-C-0001.

References

Dent, L., Boticario, J., McDermott, J, Mitchell, T., and Zabowski, D. (1992). A Personal Learning Apprentice. In *Proceedings of AAAI-92* (pp. 96-103).

Durfee, E.H. & Lesser, V.R. (1991). Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation. *IEEE Transactions on Systems, Man, and Cybernetics*,21(5),1167-1183.

Cammarata, S., McArthur, D. and Steeb, R. (1983). Strategies of Cooperation in Distributed Problem Solving. In *Proceedings of the IJCAI-83* (pp. 767-770).

Feldman, M. (1987). Electronic mail and weak ties in organizations. *Office Technology and People*,3.

Garey, M.R. & Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman and Co.

Greif, I. (1982). PCAL: A personal calendar. (Tech. Rep. TR-213). MIT Laboratory for Computer Science.

Kelley, J.F. & Chapanis, A. (1982). How professional persons keep their calendars: Implication for computerization. *Journal of Occupational Psychology*,55,141-156.

Lee, J. & Malone, T.W. (1988). How can groups communicate when they use different languages? In *Proceedings of the Conference of Office Information Systems.* Palo Alto, CA.

Raiffa, H. (1982). *The Art and Science of Negotiation.* Harvard University Press, Cambridge, Mass.

Sen, S. & Durfee, E.H. (1993). The Effects of Search Bias on Flexibility in Distributed Scheduling. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence.*

Sycara, K. (1990). Negotiation Planning: An AI Approach. *European Journal of Operational Research*.46,216-234.

Sycara, K. & Roboam, M. (1991). Intelligent Information Infrastructure for Group Decision and Negotiation Support of Concurrent Engineering. In *Proceedings of the 24th Hawaii International Conference on System Sciences.* Kailua-Kona, Hawaii.

Sycara, K., Roth, S., Sadeh, N., and Fox, M. (1991). Distributed Constraint Heuristic Search. *IEEE Transactions on System, Man, and Cybernetics*.21(6),1446-1461.