

AN AUTOMATIC MEETING SCHEDULING FOR MOBILE USERS

Behrad Assadian, Simon Case, and Fang Wang

British Telecommunications, Intelligent Systems Lab,
Post Point 12, Orion Building, Adastral Park,
Ipswich, England

Abstract. Emerging mobile computing technologies have increasingly enhanced the ability of people and organisations to work in collaboration at different locations and even on the move. By using various handheld devices such as PDAs, mobile phones, and electronic pens, remote and mobile people can flexibly exchange text and multimedia information with each other, and virtually work together as a cohesive group independent of time and location (constraints). Effective mobile collaboration requires reliable communication methods to connect people. More importantly, in-time, on-demand meeting scheduling design is necessary so as to bring fixed or mobile people together to discuss collective activities, such as planning of group work and allocation of resources whenever requested. As people usually have different availability and time arrangement preferences, quickly identifying preferable scheduling solutions for a group of people with heterogeneous and sometimes conflicting requirements is a challenge.

1 Introduction

Recent commercial and academic work on automatic meeting scheduling suggests a promising solution to handle the above challenge, by automating the process of communication and negotiation between multiple calendars. The existing automatic meeting scheduling technologies provide automation at diverse levels, ranging from simple free time display to automatic delivery of negotiated results. These technologies, however, have several limitations. First the solutions provided by these technologies are usually unsatisfactory since they lack proper capture and handling of user personal information such as preferences and requirements during scheduling. Second these technologies deploy rigid negotiation thus they cannot quickly reach solutions adaptive to varying scheduling conditions. The computational cost becomes unfeasibly large with the increased complexity and scale of scheduling. Privacy is an important issue in group coordination. Some commercial scheduling toolkits allow mutual access of user availability information. This actually prevents their use in practice, since many people are reluctant to disclose their personal time arrangement to others. An ideal meeting scheduler should be able to find the best meeting solutions fully automatically and flexibly according to different user preferences and requirements. Moreover, the architecture of the scheduler should be designed to

protect user privacy and with the consideration of the limitations of current mobile devices so as to work with the low computational power, small memory and low bandwidth problems encountered by the mobile users.

This paper presents a novel approach, *iMeeting*, to achieve automatic meeting scheduling for mobile users. This approach implements intelligent agents and fuzzy logic functions to achieve the efficient personalised and flexible scheduling. *iMeeting* quickly and automatically delivers the best available scheduling solutions for a proposed meeting, which most satisfy user preferences and requirements. Meanwhile, user privacy is well protected in *iMeeting* due to the completely automatic scheduling on a client-server architecture. *iMeeting* has a specially designed web portal for the easy access and use on small handheld devices which have limited functionality.

2 Automatic Meeting Scheduling

Meeting arrangement or scheduling is an ordinary task faced by almost everyone everyday. Though simple as it looks, meeting scheduling in practice is a time-consuming and tedious process usually involving multiple iterations of time proposing, declination, and negotiation. This process becomes even more complex and difficult, when the number of people invited for a meeting is large and when the people are distributed at different locations (or time zones) and have different preferences or bias of time management. To solve this problem, automatic meeting scheduling that automatically arranges meetings for distributed people based on various personal requirements becomes a desirable solution.

Today there are a number of commercially available software packages for scheduling meetings. Well known examples include Microsoft's Outlook, Lotus' Notes, and Meeting Maker. Some e-conferencing software such as Polycom's Conference Suite also includes scheduling mechanisms with network awareness. These schedulers usually provide an interface to view the free and busy time of the attendees of a proposed meeting. They can also search the next available time that every meeting attendee is free. The subsequent proposal of available time(s) to other meeting attendees for agreement and confirmation has to be manually executed by the meeting proposer himself, usually by sending emails. Because commercial schedulers have no consideration of user preferences or personal requirements in time search, the proposed time results are often rejected by other attendees and the proposer has to find other available times to propose. The above software therefore only provides a useful tool to assist availability comparing and searching, but they are not automatic schedulers *per se*. Frequent communication between users can make scheduling unnecessarily prolonged and frustrating. This problem would be exaggerated in mobile collaboration when quick responses cannot be obtained from mobile workers. Another important issue in current commercial scheduling software is that they have no protection of user privacy. They permit the sharing of personal calendar information by providing free or busy time for others to review. This is unacceptable for most people, especially if the calendars contain private information reflecting personal time arrangement and preferences.

In recent years there has also been intensive investigation on using agent technologies to achieve scheduling automation for distributed users [1], [2]. Endowed with certain knowledge and information about their users and relevant jobs, autonomous or semi-autonomous agents are employed to process calendars and coordinate with each other a mutually acceptable time for a meeting on behalf of their users. For coordination between distributed users, protocols ranging from simple but effective contract net to recent market mechanisms are extensively employed. In the contract net protocol [3], an agent for a user (usually the agent for the meeting host) proposes one or more time proposals according to its user's availability. The other user agents then bid for the proposals. User preferences or biases were considered in [4] when proposing and accepting time slots. User preferences, however, were values defined and input by users themselves. This obviously increases the workload of users, and more seriously, decreases the usability of the scheduling software since personal preferences or requirements are always difficult to define in exact numbers. Because the coordination and negotiation between user agents always follow a fixed routine (e.g., one agent proposes time slots and others bid, a decision on a proposed time slot is based on bids from all agents, etc.), the scheduling procedure cannot make timely responses for varied calendar conditions and user requirements. This inevitably causes ineffective scheduling which has unnecessary information collection and computation.

2.1 An Effective Solution for Meeting Scheduling

The approach proposed here for automatic meeting scheduling, called iMeeting, adopts intelligent agents and fuzzy logic to seek the best available solutions based on user personal information through flexible and efficient coordination. User personal information such as preferences for a time slot and the importance of a meeting to a user can be either input by user through natural language, or automatically learned by agents through the users' previous time-management behaviour or other exterior information proactively collected by the agents. Meanwhile, a kind of lightweight adaptive agent is designed to implement flexible and efficient coordination between user agents. Based on an easy-to-use web portal, iMeeting provides automatic meeting scheduling for users at any time and anywhere from any device. Figure 1 illustrates an overview of the iMeeting software. iMeeting develops a client-server architecture to cope with the limitations of current handheld devices whose computational power and operating systems (e.g., PalmOS) are generally incapable of running large data processing and handling tasks such as incoming request in the background. The iMeeting server stores user calendar information and undertakes necessary computation tasks for calendar management and meeting scheduling, whereas client machines provide an intuitive interface for users to request and receive all kinds of calendaring and scheduling services. All scheduling work is dealt within the iMeeting server through internal interactions between agents. Even inside the system, iMeeting agents only request information about other users from their user agents when it is really necessary. Users are only informed of automatically searched solutions (via their user agents) when scheduling is done. iMeeting avoids any unnecessary disclosure of users' personal information thus protecting user privacy to a great extent.

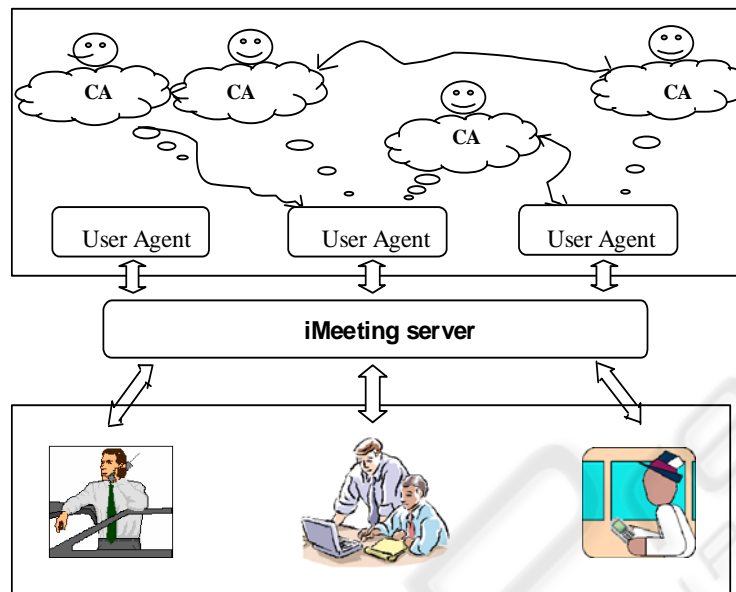


Fig. 1. Overview of the iMeeting software. Communication agents(CA)

2.2 Personalised User Assistance

iMeeting provides two kinds of intelligent agents to achieve flexible automatic meeting scheduling. At the core of iMeeting is a set of personalised agents for users. Under users' control, *user agents* act on behalf of users to process their calendars and relevant scheduling tasks based on user personal information (e.g., user preferences and requirements). iMeeting makes use of fuzzy logic to compute with words and deal with uncertainty. User personal information is represented and processed as fuzzy words [5]. In consequence users of iMeeting can easily describe their preferences or requirements by using natural language. For example, a preferred time for a meeting can be simply specified in natural language expressions such as *morning*, *afternoon*, *early morning*, *around 1:00*, etc. The information is then interpreted by iMeeting using predefined fuzzy logic functions. User personal information can also be learned by user agents from user actions during time arrangement or from exterior information such as calendar, user interest profile or organisation chart. When scheduling a meeting, user agents search and combine appropriate personal and exterior information to propose suitable time slots for a meeting. Moreover, the agents employ the information to estimate factors relevant to a proposed time slot. In particular, the agents can decide on the following features on behalf of their users: *importance of the meeting to the user*, *importance of the user to*

the meeting, and overall busyness of each user. *Importance of the meeting to the user* is a measure of how useful the meeting is to the end user. *Importance of the user to the meeting* on the other hand is a measure of how essential a given user is to the success of the meeting. A senior manager for example may well be very important for the organisers of a given meeting (i.e. they have a high *Importance of user to meeting* measure) whilst not attaching much importance to the meeting themselves (i.e. they give a low *Importance of meeting to user* value to the meeting.) *Overall busyness* is a measure of generally how busy the given user is over a time period. If a given user has a high *overall busyness* rating then they would probably prefer meetings to be scheduled at a different time if possible. The agents learn these values from responses to previous proposals combined with external information such as the position of the attendees within the organisation. Based on the information of calendar, preferences and various importances, a value between [0, 1] is automatically generated by the fuzzy system to imitate the response of a user to a proposed time slot. This value contributes to the overall fitness of a proposed time slot, which is evaluated in the agent coordination.

2.2.1 Evaluating User preferences

The evaluation agent uses busyness of user at a given slot (a crisp time and duration), importance of user to meeting, availability of user, preference of user for the proposed meeting slot, and the importance of the meeting to the user. The technology requires that a meeting proposal has the following information: time, duration, attendees. It may also include some or all of location, agenda, subject and importance of the user to the meeting. These inputs are combined in the evaluation agent in a fuzzy classifier to produce an output of the form [0,1] which is passed to the scheduler. Information exterior to calendar information such as known user interests and organisational information can be used to generate the inputs to the fuzzy classifier from the proposal. A fuzzy sub-system in the evaluation agent is used to generate the value of importance of the meeting to the user by combining and existing information on: the attendee list, organisational chart, acquaintance list the user interest profile and the agenda and/or meeting subject. A fuzzy sub-system in the evaluation agent combines information from the users calendar with the location of the meeting to decide whether a user can attend a meeting or not. A fuzzy sub-system in the evaluation agent looks at the users calendar about a given slot to allocate a busyness value to the slot. The evaluation agent adapts the fuzzy rule system when a user reschedules a meeting made on his/her behalf. The proposer agent accepts a meeting request from the user (or other agent working on behalf of the user) and generates a meeting request. The proposer agent maintains a profile of other attendees responses to meeting request. This profile is used to generate an *overall busyness* for the user. A fuzzy system within the proposer agent combines the overall busyness with the position of the user in the organisation and acquaintance profiles to assign an *importance of user to meeting* value to each attendee in the meeting.

2.3 Flexible Adaptive Coordination

When arranging a meeting, the meeting proposer sends the iMeeting server a meeting request including information on the duration, invited attendees, and various other meeting requirements if known. Examples of meeting requirements are the minimum percentage of attendees that should attend this meeting, and the *essential* attendees who must attend the meeting. The meeting request is accepted by the user agent of the proposer and then forwarded to the user agents of the meeting invitees. In contrast to other academic or commercial schedulers, iMeeting treats each attendee equally so every user agent involved in a meeting has the ability to propose its own time slots (a crisp time and duration) which are preferred by its user. To manage the proposed time slots, a set of lightweight adaptive agents are created to coordinate with each other and with user agents to find the most successful proposed time slot. These agents are called *coordination agents*. Due to the use of coordination agents, user agents are relieved from the intensive coordination work so that they can be used to provide other useful personal services for their users, e.g., calendar management or other meeting scheduling.

Coordination agents are adaptive agents that take flexible actions according to the current environments and conditions. After creation, they immediately seek other coordination agents which manage the same time slot, and combine with them into one agent in order to increase the satisfactoriness of the proposed time slot. Coordination agents make instant decisions on the feasibility of the proposed time slots based on the meeting criteria and responses currently collected from some of the user agents. Thus any inappropriate time slots can be immediately identified, and accordingly the corresponding coordination agents will be terminated. This avoids unnecessary scheduling on unsuitable solutions. The resulting scheduling can therefore be achieved quickly and efficiently with low computational cost. This property is especially beneficial when there are a large number of people involved in a meeting or the people have disparate schedules, since the increased heterogeneity and inconsistency between these people would help to remove unsuitable solutions at an early stage. Experimental simulations have proved that the computational cost of iMeeting could be at least one third less than the contract net or heuristic search, when the number of people is increased to more than 50 (assuming the calendars of these people are independent).

Before a coordination agent is terminated, the agent will inform its agreed user agents of the failure. The user agents can then consider initiating new time intervals by relaxing the constraints of their users. Due to the parallel scheduling of multiple coordination agents, iMeeting can not only promptly deliver the best available solution that most satisfies meeting requirements and user preferences, but also suggest several (2-3) alternative high quality solutions for users to choose. In particular, when a solution satisfying all user requirements is absent, iMeeting will provide several results that are likely to be accepted by the users. The users (especially the meeting host) then have the option of accepting one of the results.

2.4 Easy to use and access

iMeeting is easy to access and use. It provides a general web portal that can be accessed by a wide-range of devices ranging from lightweight devices (e.g., smart 'phones, PDAs and laptops) to desktops, workstations and enterprise servers. In particular, a simplified and succinct web interface is specifically designed to accelerate and facilitate the use of iMeeting on mobile devices. Recently a team of researchers at the University of Maryland and Microsoft created a “calendar-on-steroids” for keeping track of schedules on Pocket PCs [6]. Though this calendar provides a functional interface to plan and analyse schedules, a question is whether mobile users would accept such complex tasks on handheld devices which have a limited functionality particularly with respect to the input side of the interface. In iMeeting, we keep the interface compact and easy to understand. The relevant management and computation functions are automatically implemented by iMeeting with least human involvement.

iMeeting offers flexible scheduling that can work with varying constraints for varied user requirements. The input for a meeting proposal can simply be the basic meeting information, i.e., *time, duration, and attendees*. It can also include some or all of the other meeting concerns such as *location, agenda, and subject*. iMeeting always seeks the best available solution to meet all kinds of user requirements when scheduling a meeting.

3 Conclusion

The demand for effective mobile collaboration from an increasingly mobile work force has motivated extensive commercial and academic work on distributed automatic meeting scheduling. The iMeeting software introduced in this paper provides a new and efficient way to automatically schedule meetings for mobile users. A Survey carried out confirmed the usefulness of iMeeting as an automatic meeting scheduler. This survey investigated the attitudes and opinions of the people toward automatic meeting scheduling using iMeeting as ordinary users. Around 70% of the people deemed automatic meeting scheduling as an important and useful tool, and more than 80% people commended the easy input of the natural language and the personal consideration in scheduling as advantages of iMeeting. iMeeting is highlighted by its privacy protection which is a serious concern of many users but neglected in most commercial products. The survey also revealed worries from some involved people, who suspected automatic schedulers won't deliver solutions which are the most suitable for their own preferences.

Like many other tools, the value of iMeeting is determined by how and how often people use it. Regular update of personal calendars and frequent use of iMeeting for meeting arrangement will increase the opportunities for iMeeting to capture and learn the exact information from user behaviour.

References

- [1] L. Garrido-Luna and K. Sycara, "Towards a Totally Distributed Meeting Scheduling System", *Lecture Notes in Computer Science*, vol. 1137, 1996, pp.85-97.
- [2] S. Sen, "An Automated Distributed Meeting Scheduler", *IEEE Expert*, vol. 12, no. 4, 1997, pp.41-45.
- [3] R.G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver", *IEEE Transactions on Computers*, vol. C-29, no. 12, 1980, pp.1104-1113.
- [4] S. Sen, T. Haynes and N. Arora, "Satisfying User Preferences While Negotiating Meetings", *International Journal of Human-Computer Studies*, vol. 47, 1997, pp.407-427.
- [5] W.R. Wobcke and A.G. Sichanie, "Diary Management with Fuzzy Preferences", Proceedings of the Fifth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, 2000, pp.105-120.
- [6] B.B. Bederson, M.P. Czerwinski and G.G. Robertson, "A Fisheye Calendar Interface for PDAs: Providing Overviews for Small Displays", HCIL Tech Report #HCIL-2002-09, Computer Science Department, University of Maryland, 2002.

