PROCEEDING ROMAN 93
2nd IEEE International Workshop on PP. 430-435
Robot and Human Communication 3-5 Nov. 1993 - Tokyo

Virtual Conference Room: A Metaphor for Multi-User Real-Time Conferencing Systems

Makoto Kobayashi and Itiro Siio

IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato, Kanagawa 242, Japan
Email: mkobaya@trl.vnet.ibm.com

Abstract

We have implemented a real-time desktop conferencing system, where many users can share applications along with voice data. One of its unique features is its user interface, named the *Virtual Conference Room*. Each room, shown in a window on the PC monitor, represents a conference status, and each participant is represented as an animation character, called an *agent*. Conference management, such as floor passing, is executed through direct manipulation on agents. Voices are intermixed so that they reflect the positions and status of agents within the room.

The *Virtual Conference Room* has achieved features suitable for multi-user conferencing systems, such as visualization of the conference status, unified floor control, and dynamic subgrouping of participants.

1 Introduction

Real-time conferencing systems have been extensively studied as one of the key areas in groupware[3]. In such systems, users can synchronously share applications, such as chalkboard and text editors[10], and/or various media, such as voice and video[8][5][4].

They can be categorized into some groups based on their environements: one criteria is whether the connection is one-to-one[6] or many-to-many[1][12], and another is whether users are local (i.e., face-to-face)[10], or remote.

We are interested in remote systems, where not only a few but many people can participate in many-to-many connections, since they provide the platforms for some promising applications such as remote education systems.

In such systems, users should be allowed to dynamically join or leave the conference while it is being conducted[9].

But it is difficult, especially for the joining member, to recognize the current status of the conference, since there are many users doing various things, such as speaking and operating on shared applications. So the visualization of the conference status is one of the key issues here.

Using the motion video of each participant, as realized in MERMAID[12][7], is one way for coping with this issue, but it requires the high band-width in network and cannot be a practical solution, at least for the time being, for a conference of, say, ten or more participants.

Another issue is that the participants should be allowed to make up the subgroups within the conference so that the parallel conversations are possible. Subgrouping should be simple and dynamic, that is, without the need for creating additional conferences and moving participants into them.

In COLAB, primitive subgrouping is provided based on the page unit of the shared graphic editor[11], but still it is difficult to recognize who belongs to which subgroup.

In this paper, we will describe the system for multi-user conferencing we have implemented. Its main focus is on its user interface designed for addressing the above issues.

The user interface, which we call the *Virtual Conference Room*, aims to visualizes the conference status by simulating a conference room on a PC window, where each participant is represented as an animation character, called an *agent*.

Conference management is executed through the direct manipulation on agents. Also, participant voices are intermixed according to the positions and the status of the agents, so that the parallel conversations can be achieved.

In the following sections, we will first describe the model of our multi-user conferencing system. The details of the *Virtual Conference Room* will be shown in Section 3. The voice mixing mechanism will be described in Section 4, followed by some conlusions we have reached.

2 Multi-user Conferencing System

2.1 Conference Model

The following is the description of the model of the multi-user real-time remote conferencing system we have implemented:

- User
  Each user of the system is associated with a unique ID, or 'login name', and can log into the system from any PC where this system is installed.
- Conference
  A conference is a shared context among a set of users in the system. The context is kept consistent on every user in real-time by exchanging information through the network.
  There can be multiple conferences, independent of each other, being held simultaneously in the system. A conference is either public or private: the existence of a public conference is known to all the users, even to those who are not allowed to attend it.

- Participant
  A participant is a user who is attending a conference. One user can participate in multiple conferences at a time, although it seems rather rare that one is active, i.e., speaking or operating the shared applications, in more than one conference simultaneously. Still, it is useful to be able to monitor the status of other conferences one is not much involved in.
  A user can participate, or *join*, the conference which is being held by other participants, or *leave* the conference before it is closed. Join and leave are initiated either by the joining or leaving user, or by the request of the participant in the conference.
- Enrolled Member
  In order to be able to attend a conference, a user must be registered in the conference beforehand. A user, who is allowed to attend the conference, is called its *enrolled member*.
  A user can know the existence of conferences where he or she is enrolled or which are set open by making a query to the system.
- Chairperson
  A chairperson is one of the participants and can control the conference, such as to register a user as its enrolled member, to bring a user into it, or to close the conference itself.
  The role of the chairperson can be transfered to any other participant at any time. The user who has created a conference is initially set to its chairperson.
- Tool
  A tool is a shared application used in a conference. One example is a shared chalkboard, where partcipants can type in texts or draw graphics. Tools may or may not follow the strict WYSIWIS rules[11].
  Various kinds of tools may be used in one conference at the same time; for example, one participant is drawing on a chalkboard while another is changing a cell value in a spreadsheet.
  A tool does not always have to be shared in a conference; it can be in 'private' mode at some times. For example, a user can draw graphics in a chalkboard which is not associated with any conference, and then joins a particular conference and then brings the chalkboard into it. At that time, the copy of the chalkboard appears at each participant. Conversely, the shared chalkboard becomes private when it is pulled out of the conference. At that time, all the copies on each participant's workstation become independent of each other.
- Floor
  Some tools are designed so that they can accept multiple user inputs simultaneously, such as in GROVE[2], but other tools, especially existing applications, are not allowed to get multiple inputs. For those tools, the floor mechanism is needed, so that only one participant is allowed to type in at one time. The floor is passed from one participant to another by request, and there are several ways, called *policies*, for doing that. One way is the chairperson is to be the arbitrator; another is the system handles the request by some rules, such as the requester gets waited until the current floor holder explicitly releases it, or it is immediately passed to the requester.
  Different policies should be allowed for different tools used in the conference, so it is important that participants do not get confused at which tool is using which policy.

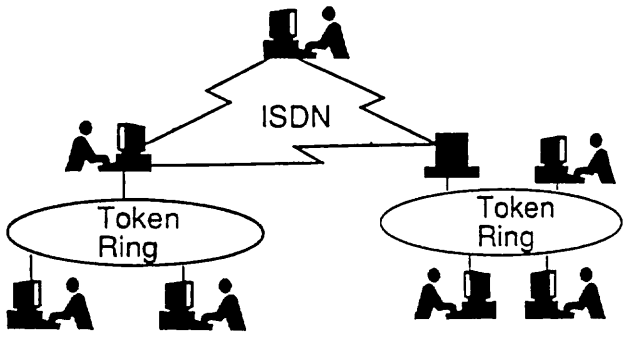

Figure 1: Typical Network Configuration

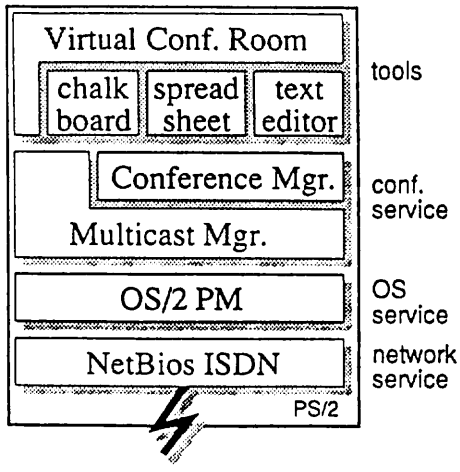

Figure 2: System Structure

2.2 System Structure

Figure 1 shows the typical network configuration in which this system runs. Remote sites are connected through 64 kbps ISDN lines.

The system consists of the following four components:

- Multicast manager
- Conference manager
- *Virtual Conference Room* module, and
- Tools.

The total structure of the system is shown in Figure 2.

The multicast manager enables broadcasting of data packets among the PCs belonging to the same conference. Also, it takes care of the packet consistency when join or leave of a participant occurs.

The conference manager sends the join or leave notice to tools and the *Virtual Conference Room* module when such an event occurs. The floor passing control is also handled by the conference manager.

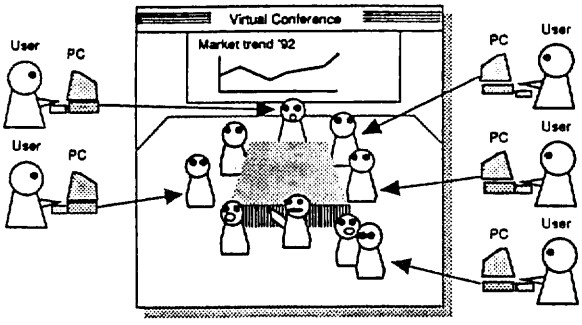

Figure 3: Virtual Conference Room

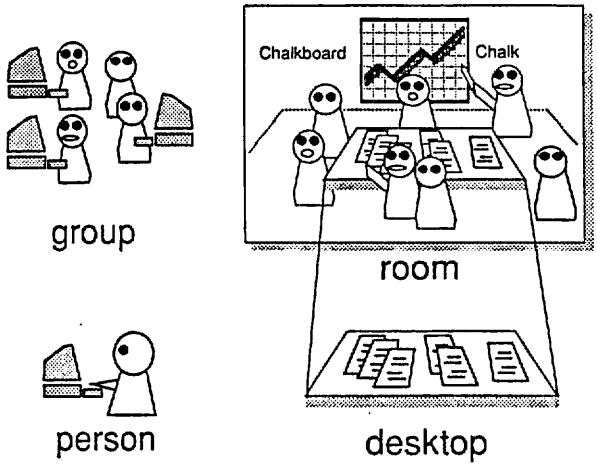

Figure 4: Desktop and Room Metaphor

The *Virtual Conference Room* module gives the user interface for the whole system, and will be described in detail in the next section. Tools and the *Virtual Conference Room* module uses the services given by the two managers.

3 Virtual Conference Room

The *Virtual Conference Room* aims to provide the visual and interactive environment to the users of this system. A conference itself is represented as a room in a window on the PC monitor, and each participant is shown as an animatoin character, called an *agent*, in the room as shown in Figure 3.

The view is synchronized among all the participants, and the conference control, such as floor passing, is done through direct manipulation on the agents.

The *Virtual Conference Room* is based on the *room metaphor*, which is a natural extension of the desktop metaphor for the multiple user environment (Figure 4). It is consisted of the following elements:

- Agent,
- Conference Room, and
- Conference Items.

Each will be described in some detail below.

3.1 Agent

An agent is an animating object placed in a conference room window and represents its associated user. Each user can only manipulate his or her own agent. In the current implementation, an agent is consisted of the face and the body objects, and the face object holds various facial expression bitmaps generated from the video capturing.

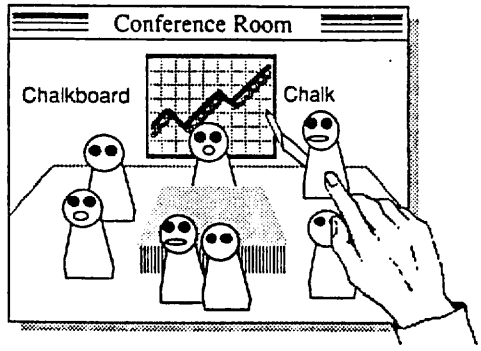

Figure 5: Floor Request

Following actions are defined for an agent:

- Speak
  The voice input drives the associated agent to change its facial expressions so that it looks as if it is actually speaking.
  This is quite useful for recognizing who is speaking, compared with usual telephone conferencing where not all participants' voices are familiar with each other. It is especially useful when more than one participant are speaking at the same time.
- Walk
  The user can drag the agent within the room. At that time, the feet are moved so that it looks as if it is actually walking in the room. So the other participants can easily notice that the agent is changing its position right now.
  The feet come out only when the agent is dragged so that they do not occupy the space in the normal state.
  As will be described later, the agent position affects the voice mixture.
- Raise the hand
  By pressing the mouse button on the body, the hand is raised. This provides an easy way for drawing attensions of other participants. It is also useful for taking votes on some topics being discussed in the conference.
  As above, the hand appears only when it is raised. The voice mixing is also affected while the hand is being raised.
- Move the head
  By menu commands or key inputs, the face object can take various actions, such as nodding and turning to right and left.
- Hold an item
  An agent can hold an item by double-clicking on it. As described below, an item is usually associated with some conference control, such as floor requesting. For example, by grasping the chalk attached to a chalkboard, the user obtains the floor for using it, as shown in Figure 5.

3.2 Conference Room

A conference room is a window representing one of the conferences currently being held on the system. It contains agents and some conference items, which are described below.

Each room is associated with a set of enrolled members, who are allowed to attend it. Only the current chairperson of the conference can change the members.

3.3 Conference Items

When many users are participating in a conference, there will be more chances that multiple users try to allocate the same resource at the same time. The typical example is requesting the floor on the chalkboard. *Conference items* provide the visual clue for such control through direct manipulation on them.

They are also used for visualizing some services, such as file distributions. Three items are predefined by the system, as described below:

- Table
  A table is usually placed in the center of the conference room, and document icons placed on it represents that those documents are shared in the conference.
  When a file icon is dragged on the table, the file is distributed to all the participants.
- Stage
  A stage represents a place for explicitly making a speech. The voice gets loud as long as the agent is on the stage; the distance from the listner's agent does not affect it.
  The stage follows the mutual exclusion rule; only one agent can stay on it at a time. When more than one agent try to get on it simultaneously, only one is allowed and others will stay beside the stage making a line.
  The exclusion rule can be changed by the chairperson so that, for example, up to three agents can get on it at the same time.
- Chalk
  A chalk represents the floor for the chalkboard, and the floor request is done by double-clicking on the chalk. When more than one participant try to get hold of the chalk, the conference manager arbitrates the requests, and informs the *Virtual Conference Room* and the chalkboard that who will get the floor and who will wait, according to the policy associated with the floor. The *Virtual Conference Room* moves the requesters beside the chalk holder to form a waiting line.
  When a tool allows multiple user inputs, but still wants to set some limit on the simultaneous operations, the tool can request for the *Virtual Conference Room* to show, for example, three chalks attached to it.

By the above mechanisms, users can easily recognize not only who are holding the floors of which tools, but also who are waiting for those floors.

When the request is postponed, users can simply cancel the request by moving their agents out of the wating lines.

Binding of a floor to its item is done at run time, so users can change the representation by providing other item icons.

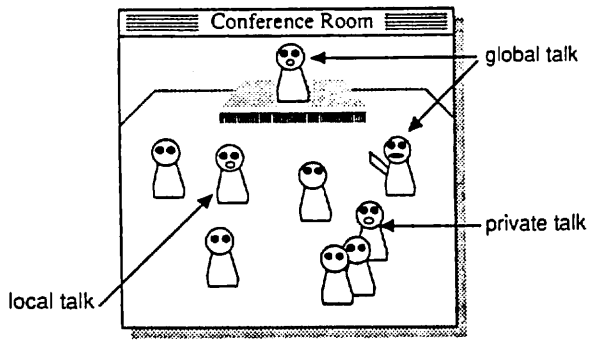

Figure 6: Voice Modes

4 Voice Control

4.1 Talk Mode

The *Virtual Conference Room* also aims to provide the natural and effective voice environment for each participant, by intermixing the voices from the other participants accorging to the following rules. Figure 6 shows three talking modes we have defined:

- Local Talk
  When an agent is in a normal state, its owner's voice is heard as in an actual room: the voice gets weaker when the listner's agent is more distant from the agent, and the relative direction is reflected in the stereo sound. When the agent walks, its voice is adjusted to reflect the new position.
  This mode is useful for making subgroups, as it is done simply by dragging agents into some groups.
- Global Talk
  When an agent is in this mode, the voice gets loud and is not affected by the distance from the listner. This simulates using a microphone in the room.
  An agent gets into this mode either by raising the hand or by getting on the stage. The voice is kept loud as long as the hand is up or the agent stays on the stage.
- Private Talk
  When two agents are overlapped, their voices are heard only within them. Other participants can notice that they are 'whispering' with each other, and can join the group by overlapping his or her own agent with them.
  This mode is useful for making 'closed' subgroups.

4.2 Mixing Methods

We have implemented the voice mixing mechanism in the following two ways:

- Centralized mixing
  A server PC is provided with the voice mixing hardware attached to it. Each user's voice is put into this hardware, which is an n-by-n register array, and the controller sets the parameters for each of the par-

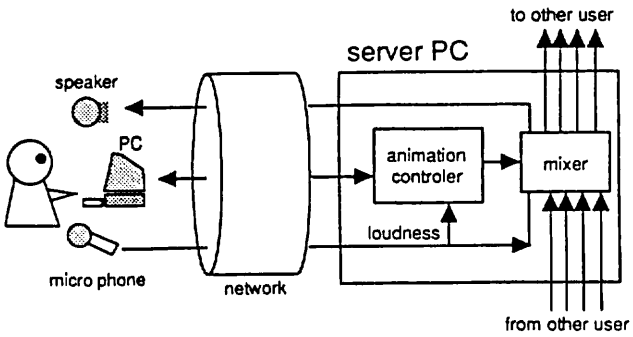

Figure 7: Voice Control Mechanism

ticipant according to the positions and status of the agents in the conference room. Figure 7 shows the schematic view of this approach.
This approach is suitable for using telephone lines, since the voice can go through separate lines with application data. But it has a drawback that the number of participants is limited by the input channels of the mixing hardware.

- Distributed mixing
  The voice is packetized and sent to all the participants, as with other application data. Each participant receives voice packets of the other partcipants and the mixing is done locally, either by the CPU or on the voice I/O card.
  The merit of this approach is that there is logically no limit on the number of participants, but the deficiency is that there sometimes occur discontinuation in voice because of the time jitter on the packet delivery over the LAN.

5 Conclusions

We have implemented a multi-user real-time conferencing system with the unique user interface, called the *Virtual Conference Room* , on PS/2 running OS/2. The multicating manager and the conference manager are implented in C and the *Virtual Conference Room* is implemented in Smalltalk/V PM.

Figure 8 shows the actual screen image of the *Virtual Conference Room* . There are five users participating in the conference, and three users are using the realistic face image obtained from the video capturing, and the other two are using drawn pictures. The chalkboard is shown as an icon on the back wall, with a piece of chalk attached to the right side. The actual chalkboard is represented in another window outside the conference room.

Using the *Virtual Conference Room* as the user interface, we could achieve the following features suitable for real-time conferencing systems where many users attend at the same time:

- Visualization of the conference status
  Participants can easily recognize the current status of the conference, such as who is speaking and who is operating on which tools. This is especially useful for a user jumping into the conference.
- Unified floor control
  Easy floor control is important since there are more chances of request collisions. Conference items give the unified way of floor requesting independent of the tools being used in the conference. They also provide the same operation for the request canceling.
- Dynamic subgrouping
  Participants can dynamically set up subgroups and make parallel conversations without interfering with each other, simply by dragging their agents in the room. It is also helpful that who belongs to which subgroup is easily recognized.

6 Acknowledgment

The authors wish to thank A. Nakajima, F. Ando, and T. Sakairi for their collaboration in designing and implementing the conferencing system. They also wish to thank T. Kurosawa for giving them chance to work on this research.

References

[1] S. R. Ahuja, J. R. Ensor, and D. N. Horn. The Rapport Multimedia Conferencing System. In *ACM Conference on Office Information Systems*, pages 1–8, March 1988.

[2] C. A. Ellis and S. J. Gibbs. Concurrency Control in Groupware Systems. In *Proceedings of ACM SIGMOD '89 Conference on Management of Data*, pages 399–407, Seattle, May 1989.

[3] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware - Some Issues and Experiences. *Communications of ACM*, 34(1):39–58, 1 1991.

[4] R. Fish and et al. Evaluating Video as a Technology for Informal Communication. In *CHI '92*, pages 37–48, May 1992.

[5] H. Ishii. Design of Team WorkStation: A Realtime Shared Workspace Fusing Desktops and Computer Screens. In *Proceedings of the IFIP WG 8.4 Conference on Multi-User Interface and Applications*, Heraklion, Greece, Sept. 1990.

[6] H. Ishii and M. Kobayashi. ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. In *CHI '92*, pages 525–532, May 1992.

[7] T. Ohmori and et al. Distributed Cooperative Control for Sharing Applications Based on Multiparty and Multimedia Desktop Conferencing System: MERMAID. In *12th Int. Conf. on Dist'd Comp. Sys.*, pages 538–546, June 1992.

[8] R. W. Root. Design of a Multi-Media Vehicle for Social Browsing. In *CSCW '88*, pages 25–38, Portland, OR, Sept. 1988.

[9] S. Sarin and I. Greif. Computer-Based Real-Time Conferencing Systems. *IEEE Computer*, 30(1):33–45, Oct. 1985.

[10] M. Stefik and et al. Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1):32–47, 1 1987.

Figure 8: Screen Capture of the *Virtual Conference Room*

[11] M. Stefik and et al. WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *ACM Transactions on Office Information Systems*, 5(2):147–167, 4 1987.

[12] K. Watabe and et al. A Distributed Multiparty Desktop Conferencing System and Its Architecture. In *IEEE Phoenix Conference on Computers and Communications*, pages 386–393, Phoenix, AZ, Mar. 1990.