Distributed Research Teams: Meeting Asynchronously in Virtual Space

Lia Adams, Lori Toomey, and Elizabeth Churchill
FX Palo Alto Laboratory, Inc.; 3400 Hillview Ave., Bldg. 4; Palo Alto, CA 94304
{adams, toomey, churchill}@pal.xerox.com

Abstract

As computer networks improve, more social and work interactions are carried out "virtually" by geographically separated group members. In this paper we discuss the design of a tool, PAVE, to support remote work interactions among colleagues in different time zones. PAVE extends a 2D graphical MOO and supports synchronous and asynchronous interactions. PAVE logs and indexes activities in the space. This capture facility enables playback and augmentation of meeting interactions by non-collocated group members. Thus, members can participate *asynchronously in meetings they could not attend in real time, not just review them.*

1. Introduction

In recent years, improvements in computer hardware, software and network infrastructures have increased digitally based interactions between geographically separated individuals. Interactions between members of these virtual communities include socializing, game playing, information sharing, sales and advertising, leisure pursuits, and collaborative work.

These on-going developments have created new working group possibilities and the emergence of digitally based "virtual communities" [12, 13]. In this paper we consider the design of tools that support such distributed work interactions. More specifically we discuss the design of PAVE, a tool developed at FX Palo Alto Laboratory to support research meetings between non-collocated research team members. Our research focus has been to support both synchronous and asynchronous activities, enabling spatially and temporally separated collaborators to work together on projects. In the next sections we describe the working group activities we aim to support, and we review existing tools for distant collaboration. Then we describe the design of PAVE and illustrate its use with a detailed description of a meeting scenario.

2. Research Meetings

The focus of our research is on the support of small research team group meetings. Within these meetings, the primary function is to share information about project-related activities; this information sharing enables members to make decisions about project actions and goals. The groups in question have a heterarchical structure, and information flow tends to follow a connected network communication structure.

Ordinarily, such meetings occur in real time between collocated colleagues. It is clear that physical proximity enables fine-grained postural, gestural and speech interactions, and the sharing of artifacts. It also offers a natural frame for interjection, graphical and anecdotal illustration, and thus for participation in decision-making processes. Physical proximity therefore also supports feelings of participation and group membership.

Increasingly, however, our research meetings are made up of geographically separated colleagues who work in different time zones. Clearly, being geographically separated can interfere with spontaneous interjections and with sharing documents and other artifacts; group members are unable to easily follow discussion on a topic without aid of tools like telephone conferencing and video-conferencing. When the group members must work asynchronously, real-time involvement in discussion is not possible. Thus, when asynchronous collaboration is the only option, but participation is desired, the challenge is to provide sufficient context and detail of the decision-making processes to inform group members and enable involvement and meaningful comment. Much work has already been done to address the problems of informing and involving distributed co-workers, as we summarize below.

3. Requirements and Related Work

In order to support our target group activity, i.e., meetings of non-collocated colleagues in different time zones, we need elements from the domains of real-time collaboration support, asynchronous groupware, and meeting capture and replay.

3.1. Supporting synchronous work

A number of tools provide awareness of the activities of other (remote) group members and enable real-time conversation and activity sharing, e.g., videoconferencing and desktop conferencing systems, media spaces, chat systems, and virtual spaces [1]. The expense and overhead of videoconferencing generally limits use of such systems to infrequent, formally scheduled events. Desktop conferencing systems support more lightweight interactions, and feature text and/or audio (and occasionally video) for communication, mechanisms for awareness, and shared tools (e.g., whiteboards and text editors) for creating and manipulating artifacts.

Some desktop conferencing systems, such as TeamRooms [26] and its successor TeamWave [29], use a "place" metaphor (i.e., there are various rooms in which people meet, some designed for specific purposes, and the artifacts within those rooms are persistent across sessions). TeamRooms is primarily a text-based system, with graphic additions. Participants can communicate textually, share views of documents, and leave notes for one another. The latter is the only form of asynchronous communication. TeamWave provides the infrastructure for on-line interaction, supplemented with tools such as electronic whiteboards that mimic the tools collocated workers have at their disposal. In essence, TeamRooms tries to create a digital version of the traditional office with its opportunities and affordances for both formal and informal communication.

Other environments, such as the RoundTable product from ForeFront [27], use a meeting metaphor, and are geared specifically towards supporting formal meetings among distributed participants. In addition to text-based chat, participants may share images, documents, World Wide Web locations, audio, and video. Rather than gather together physically, each user connects to a server from his own office at an agreed upon time.

Providing a different twist on on-line meetings are systems such as DOLPHIN [19] and the Electronic Meeting Room (EMR) at the University of Hawaii, Manoa [9]. These systems are based in physical meeting rooms, in which participants contribute via a keyboard and/or pen-based interface. Proponents claim that meetings are shorter because participants are able to work in parallel, and "talk" (type) simultaneously.

Chat systems and place-based virtual spaces known as MUDs (Multi-User Dimensions) and MOOs (MUDs Object-Oriented) were originally created for social purposes and games [3]. In simple chat systems, users converse with one another by typing text. MOOs and MUDs also feature textual conversations, but these exchanges occur in a place-based context, and users can extend the space and create artifacts that define a particular place (e.g., creating virtual books and tables for a library room).

More recently, MUDs and MOOs such as The Palace [22] have become graphical spaces, augmenting text chat by the immensely popular use of avatars (a visual representation such as a graphic or photo of oneself) and other images or props. Still, the focus has remained primarily social. Notable exceptions are the Jupiter system from Xerox PARC [4], Jupiter's successor PlaceWare [24], and Electric Communities [8]. Jupiter supplemented a text MOO with audio, video and interactive artifacts in order to make it a richer environment for professional work.

The popularity and sense of community created by these spaces suggest that some of their features (e.g., the place metaphor and the use of avatars) may be important elements in creating an inviting environment in which participants engage in collaborative work.

Excluding video-conferencing, all of these systems share some advantages in addition to just supporting communication between (possibly) non-collocated users. Because interaction takes place textually, it is simple to produce a written record of the proceedings. It is also possible for participants to remain anonymous if they wish. Finally, because these systems have a much lower overhead for participants versus traditional video-conferencing, they are practical for a variety of daily interactions.

These systems also share some shortcomings. Despite the claims of some desktop and video-conferencing systems, none of these technologies provides the broad spectrum of communication possible in face-to-face meetings. A key ingredient missing is the ability to transmit and sense cues such as turn-taking and acknowledgment of the floor that are often conveyed with body language and/or subtle facial expressions. These conventions are further hampered by the latency created by the input of textual utterances. Another drawback is in the maturity of the available tools to support work, e.g., shared editors, whiteboards, drawing tools, etc. Rarely do the tools provided in these systems approach the sophistication we readily find in their single user counterparts. Ideally, we would like to be able to incorporate the tools users are accustomed to having available on their desktop into these shared spaces. Finally, with the exception of leaving behind notes or other shared artifacts, these systems are still limited to synchronous interaction.

3.2. Supporting asynchronous group work

Because many kinds of workplace communication cannot be done synchronously, groupware and other tools for computer-supported cooperative work have been developed for supporting asynchronous workplace activities. Tools and suites such as Lotus Notes [17] provide strong support for asynchronous communication and coordination.

The most common tools for asynchronous communication are electronic mail, newsgroups, and groupware tools like Lotus notes. These tools not only allow a single user to dispatch information to a large number of people, but also support multi-user discussion such as brainstorming and other cooperative activity. Such systems preserve documents but do not often preserve the context in which those documents were created.

Both email and newsgroups create new documents at every communication; an extended discussion in either medium depends heavily on extensive quoting of earlier notes, to provide context for new additions. Other groupware tools focus on creating, updating, and sharing persistent documents. When co-workers collaborate asynchronously on a shared document, a coordination mechanism is needed to alert them to new work-related documents and versions, and to keep one user's updates from interfering with another's, whether they are made in disjoint or overlapping periods of time. Systems that do version control, such as RCS [30] for program source code, and Documentum [5] and DocuShare [6] for general documents, are common examples of coordination mechanisms. Other CSCW tools that manage documents may use a version-control system as part of their infrastructure, even if the end-user does not explicitly do version control operations.

More recently, asynchronous tools are incorporating features popular in synchronous tools, such as awareness, and integration of graphical media and audio and video. Awareness in particular is a valuable addition to synchronous collaboration, since awareness of co-workers increases the likelihood of collaborating with them [25]. The Piazza System [14] embeds awareness of collaborators into artifacts themselves, with pictorial icons attached to document viewers, for example. While Piazza provides awareness of synchronous use, the idea of attaching awareness to artifacts was extended to asynchronous use in Timewarp [7], where the association of collaborators with the artifacts they have manipulated can be reviewed through a historical lens. Edwards and Mynatt describe the Timewarp system (which combines versioning and time-based browsing with asynchronous, coordinated resource sharing) as neither synchronous nor asynchronous collaboration; they call this hybrid model autonomous collaboration.

3.2.1 Replay. An emerging area in asynchronous computer-supported cooperative work is replay. Xerox PARC's meeting capture and salvage work [21] takes a real-time, synchronous meeting, captures the audio, the text recorded by a human note-taker, and drawings on the Tivoli [23] whiteboard, and saves them so that they can be revisited later. Any of the media can be replayed in coordination with the others. From a synchronous group activity, they produce a multi-media artifact that can later be reviewed, browsed, or studied asynchronously. Both PARC's WhereWereWe architecture [20] and work by Manohar [18] support this kind of multimedia replay.

3.3. Summary of requirements

Thus, to support virtual meetings between spatially and temporally separated colleagues, we require functionality drawn from the domains of real-time communication and collaboration, asynchronous groupware tools, and meeting capture. We require that our system be low-overhead to use, and that it support replay of the various media used in a meeting. In addition to straightforward replay, we need to be able to reconstruct decision-making processes and the context in which they occurred. This reconstruction should address the purpose of a meeting, namely to generate *understanding* of decisions, not just to present a list of items decided upon.

To apply the strengths of asynchronous collaborative tools to our virtual space, we must ensure that the tools that users ordinarily depend on to create and access persistent work documents are still available to them. The virtual space cannot effectively supplant these tools; instead, our goal is for the virtual space to coordinate other meeting documents. In common with real-time collaboration, we need the features of awareness and the ability to create and manipulate shared artifacts.

We hope that meetings in virtual space will share the benefits of other technology-mediated meetings, in promoting more equal participation by members [10].

4. PAVE: A virtual meeting room

In this section we describe the design and use scenario of PAVE (**PA**L **V**irtual **E**nvironment), a system intended to address the requirements specified above. PAVE is implemented on The Palace, a 2D graphical MOO with a client-server architecture. A client-server architecture was required for our prototype, so that we could *instrument* the central server, modifying it to capture all events from all clients and assign them a sequential ordering.

Our space, PAVE, is a 2-dimensional graphical MOO, structured as a collection of rooms connected by virtual doorways. A room in PAVE consists of a set of background graphics, in front of which users appear as customizable avatars that can carry props (graphical objects). Rooms and users can each contain event-driven scripts activated by movement to a spot in a room, in response to utterances, in response to arrivals and departures, or through other events. Scripts can play sound effects, add props to a room, execute painting commands, and drive a user's web browser to a particular URL.

Avatars are simple GIF images; users can switch rapidly between many different avatars to change expression. All visitors to the same room can communicate textually with one another and can draw in the same space; each of them sees the same third- person viewpoint of the room, the contents, and the other people. Thus, the room constitutes a shared conversational space [15]. Users can play prerecorded sounds but cannot converse using live audio.

To share documents and other non-conversational information, users can exchange information through client-side non-WYSIWIS (What-you-see-is-what-I-see) web browsers.

Figure 1: The meeting begins.

In addition to these features for synchronous communication that PAVE inherits from The Palace, PAVE's server is instrumented to be able to capture all events that occur: text chat, artifact creation, movement, avatar changes, drawing, and document sharing. PAVE's server also permits users to issue WYSIWIS browser-loading commands.

4.1. Synchronous meeting

We describe a scenario for using PAVE to conduct an asynchronous meeting within a distributed work group. Consider a project team, of which some members are located in Palo Alto and others are located in Tokyo, who will meet in virtual space to conduct a weekly project status meeting. The meeting will occur in two phases, as described below. The participants have an agenda for the meeting in the form of an HTML page. The California participants start the meeting at 10 AM Pacific Time, which is 2 AM in Tokyo.

They conduct their part of the meeting, sharing their status items, marking up a diagram under discussion, and updating the agenda to include new items or clarifications. Once the live (synchronous) part of their meeting ends, it has been captured and saved by the PAVE server. The participants then move to the patio in the virtual space to continue informal discussion that is not part of the meeting, and thus, is not recorded. They are now available for periodic informal exchanges throughout the day, or for participation in other captured meetings.

Figure 1 shows the virtual meeting room with the two California participants. As the images show, the room is a photograph of an actual meeting room familiar to all the group members. Research has indicated that using visual and spatial metaphors not only supports the use of appropriate activities but can aid a sense of "being there" [2, 28]. A primary function of metaphor is to provide a partial understanding of one kind of experience in terms of another kind of experience, providing a common ground for understanding situations and for selecting appropriate actions, thus offering a shared scaffold for activities [16].

There are a number of props in the room with obvious physical analogues. Group members are represented as static photographic avatars; these avatars are static gesturally, but may be moved anywhere in the represented room. To move avatars, group members can simply point and click where they would like their avatar to be moved. There are certain "hot spots" in the conference room; when an avatar is placed on these spots it is automatically transported into another room. In PAVE, there is a separate whiteboard room that a user can enter by positioning their avatar over the picture of the whiteboard at the edge of the meeting room. Here, all group members can draw illustrative diagrams. Because such items are persistent in this space, it is possible for group members attending later to see and annotate anything on the whiteboard.

Participants communicate by typing text into the gray text area along the bottom of the screen. When a person types text into the box, the text appears as speech in the form of a cartoon balloon, as shown in Figure 1. The participants are thus able to discuss issues in real time. Since long utterances create large balloons that remain on the screen a long time (potentially blocking part of the scene), we have noticed that frequent users get in the habit of breaking long remarks into smaller segments by introducing carriage returns. The size of the font used in balloons, as well as the length of time a balloon remains on the screen, is configurable by the user. If a user misses seeing a balloon in real time, or doesn't notice which utterance was made before another, PAVE keeps a complete scrollable log of all dialogue.

In addition, any participant can give commands from within the virtual space to bring up web pages in all present participants' browsers. If a group holding a meeting finds it too chaotic for all participants to have this ability, these commands can easily be restricted to the facilitator(s) only.

and in annotation. The system itself takes the role of recorder, logging all activities. During the meeting, a number of documents are introduced on topics of interest to the research group. When these are introduced, they can be seen by all participants in a separate browser that appears next to the conference room window. By launching web pages in this way, group members can share information they have produced or external documents that are deemed relevant.

The two California group members go through the agenda, discussing each topic, and updating the agenda as they proceed. For example, one agenda item is to discuss an upcoming conference; the two add a question to the agenda asking which members of the group want to attend.

When they finish their meeting, they conclude their

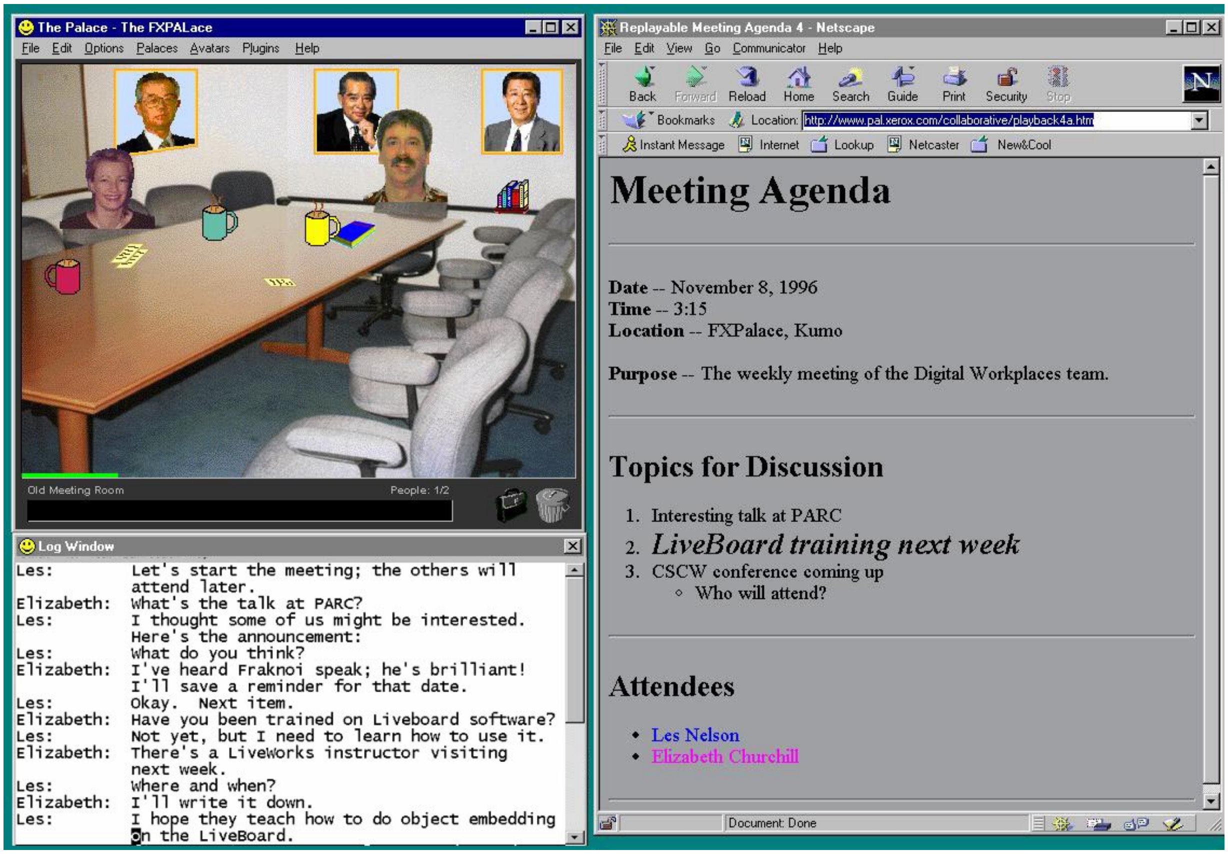


Figure 2: Screen shot showing PAVE, text log, web browser with meeting document

Figure 2 is a typical screen shot, showing the PAVE window, the text log window below it, and a separate web browser loaded with the agenda for the meeting. To conduct the meeting, one person takes the role of facilitator, stepping through the agenda items. The agenda is used by the facilitator to structure the activities but also offers a structure for later participants to follow in replay

PAVE session, and the capture process terminates.

4.2. Asynchronous meeting

Several hours later, when the members of the project team in Japan come in to work, they gather in the virtual space and commence the second phase of the asynchronous meeting. At 1 PM Tokyo time, the members open the captured meeting and begin their participation.

Figure 3: The second phase of the meeting.

The new participants add their names to the existing list of attendees on the agenda web page. After they give a command to commence replay, each line of dialogue typed in the earlier meeting is replayed as a dialogue balloon. Avatars of the original speakers appear in the positions where they first spoke each line, but their avatars appear in black and white to distinguish them from the currently live participants. Figure 3 shows this new phase of the meeting, with two color avatars of the Tokyo participants, and two black-and-white avatars of the previous participants.

The participants in Tokyo review the remarks made during the earlier phase of the meeting, adding their own remarks, possibly asking or answering questions, and updating the agenda. The Tokyo participants can interject remarks at any time during the replay; each remark is added to the log, along with information on when it occurred.

As documents are introduced or updated in the earlier version of the meeting, the new participants' browsers display them. For example, in this scenario, the agenda is updated to show the current item (using change of typeface, see Figure 4), to clarify old items (the time and place of a training class), and to show new information (such as the names of later participants).

4.2.1 Effectiveness. Instead of merely seeing a summary of the meeting held in California, with a list of decisions arrived at during the meeting, the Tokyo participants can replay the full meeting, see how those decisions were arrived at, and add their own comments and information.

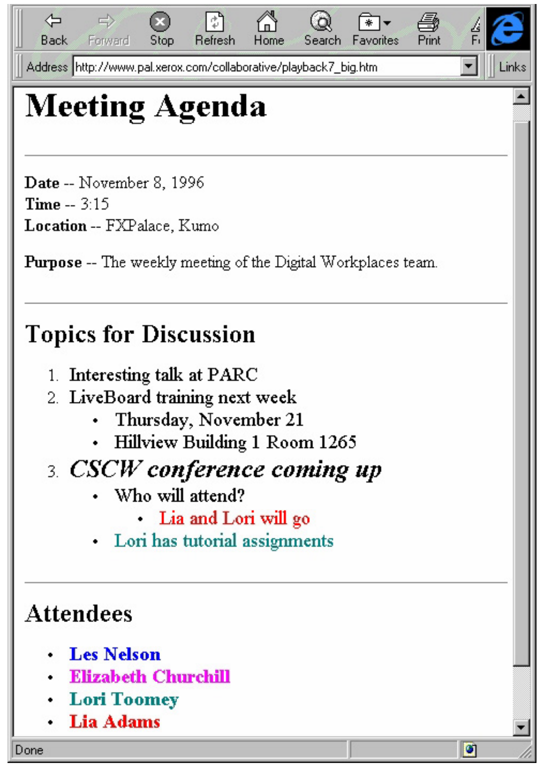


Figure 4: The updated agenda.

They can talk and respond not only to the previous captured remarks, but also to one another in their joint replay session. Any participant may issue a command to continue replay, but as is the case with browser-loading commands, this ability can be restricted to a moderator if chaos results. The replay and augmentation of the meeting is a collaborative activity, just as the earlier phase was. The contributions of each participant are seen in the context of the project meeting.

People who revisit the meeting after this second phase will see the entire composite meeting, including black and white avatars for all previous participants, and color avatars for those new participants meeting in real time. (Avatars can be shown with session numbers attached, to make obvious which previous participants were present at the same time.) The composite meeting scene shows all the participants present *as a team*. Thus, the meeting representation, while allowing viewers to determine who was present when, presents each participant as a team member, rather than distinguishing people as original contributors versus responders.

4.2.1. Implementation. As mentioned above, instrumenting the server to capture all events in the virtual space is the key to supporting asynchronous meetings.

All events are time-stamped, numbered, and associated with a session.

The time-stamp captures the actual date and time of the event, while the event number indicates position in the sequence of events. In the initial session, event numbers are integers beginning with 1. During subsequent playback sessions, event numbers are assigned using a hierarchical scheme (e.g., 1.1) to indicate insertion between events from earlier sessions. For replay views (discussed in the next sections), users specify which ordering of events they want to use. The real clock time-stamps provide an absolute chronological order, while the event numbers provide a "logical" order that interweaves events from multiple sessions.

The capture of external events including document update, which is built on a version control facility, enables decisions to be unraveled to note all disagreements and how resolution was achieved.

Participants at each site can continue a meeting in later phases, by using tools that report when a meeting contains new content and directing them to the new material without having to replay previously-seen material. (This feature is a familiar one in groupware and newsgroup tools, and will be essential for communication through asynchronous augmentation.)

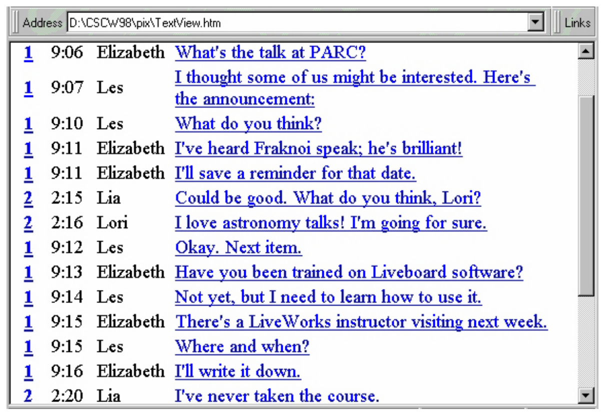
4.3. Accessing aspects of the meeting

We offer participants a number of ways of accessing important parts of the meeting. These are the Replay view, the Text view and the MultiView.

4.3.1 The Replay view. The asynchronous meeting scenario described above uses the Replay view, consisting of the replay and annotation of the whole meeting by stepping through each scene. The Replay view includes the meeting room in PAVE, the text log, and the web browser documents (displayed in a separate web browser). In the Replay view, later participants view the dialogue in its original order, interjecting their own remarks, hearing and playing sound cues, seeing and adding to drawings, viewing and updating auxiliary documents as they were introduced into the meeting, and creating new documents at any point in the replayed time-stream. A simple time-line is shown at the bottom of the PAVE window showing how much of the meeting has already been played back. In Figure 3, this time line is shown in green.

The Replay view offers an opportunity for non-collocated group members to participate in a meeting with coworkers, experiencing the same state of the virtual meeting room, the audio cues, and shared documents, as they would have if they had attended the meeting synchronously with their coworkers. By playing back events using the logical order or time-stamps, all remarks are seen in their original context.

4.3.2. Text view. While the Replay view offers a full presentation of a meeting, the Text view is designed for quicker perusal. In the Text view, the log of all text uttered in the virtual space is the primary presentation of the meeting contents. For each remark made or sound recorded in the log, the Text view shows the speaker, the time-stamp, and a time-line number showing in which thread the remark was made. (A new meeting happening on-line defines thread 1; each subsequent visit, by any group member or members, defines the next higher thread number.)



1	9:06	Elizabeth	What's the talk at PARC?
1	9:07	Les	I thought some of us might be interested. Here's the announcement:
1	9:10	Les	What do you think?
1	9:11	Elizabeth	I've heard Fraknoi speak; he's brilliant!
1	9:11	Elizabeth	I'll save a reminder for that date.
2	2:15	Lia	Could be good. What do you think, Lori?
2	2:16	Lori	I love astronomy talks! I'm going for sure.
1	9:12	Les	Okay. Next item.
1	9:13	Elizabeth	Have you been trained on Liveboard software?
1	9:14	Les	Not yet, but I need to learn how to use it.
1	9:15	Elizabeth	There's a LiveWorks instructor visiting next week.
1	9:15	Les	Where and when?
1	9:16	Elizabeth	I'll write it down.
2	2:20	Lia	I've never taken the course.

Figure 5: Text view of the asynchronous meeting.

The Text view supports rapid scan of the text log, including forward and backward search for keywords. The Text view offers a quick way to peruse the text by skimming and navigation through keyword search and scrolling through the meeting.

If the user sees something of interest, they can stop skimming and start participating. From any line in the Text view, the reader can follow links to the other views of the meeting. The views appear in the state they were in at the time the line of text was uttered. From this view, the viewer can navigate from the textual content of the meeting (the words of the participants) to the context: what words had been already said, awareness of who was present, appearance of all avatars, and the state of shared artifacts. Thus, the Text view presents a fairly compact representation of a meeting, while allowing ready navigation to richer media.

4.3.3 MultiView; threaded activity lines. Finally, we are developing a viewpoint that shows several graphical views onto the meeting activities; we call this the MultiView. This MultiView appears in a separate web page as a scrollable web frame. A prototype multi-view interface is shown in Figure 6. This interface is similar to that offered by Ginsberg and Ahuja [11]. In their interface they offer a visual history or record of distributed multi media

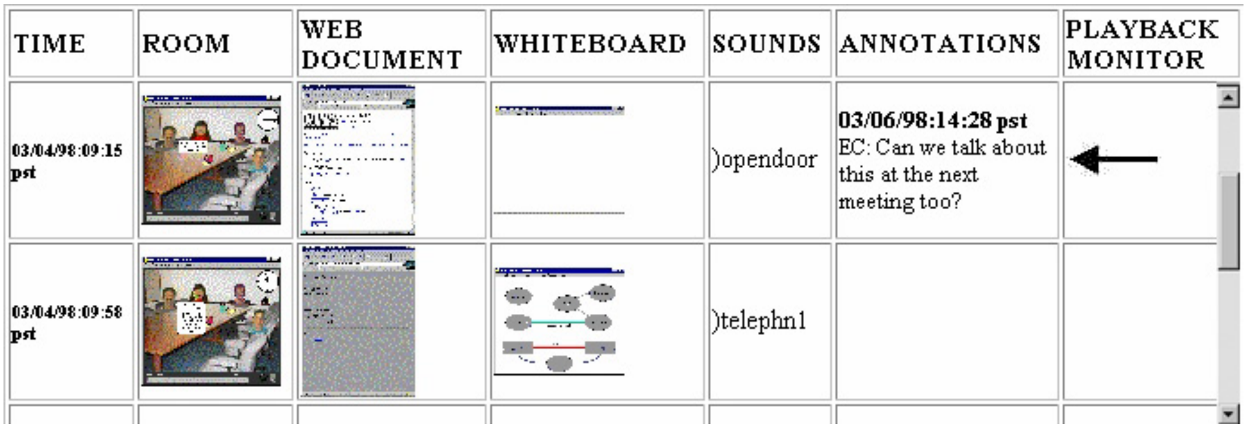


Figure 6: Prototype of MultiView.

collaborations by showing meta-information visualizations of virtual meetings.

Our MultiView meeting reconstruction interface utilizes information about the activities of “live” participants by considering activity types going through a central server. The MultiView is then automatically created. The MultiView interface shows four contemporaneous meeting cues in rows, indexed by a time-stamp. Rows are created on the basis of when events occur: the events are text speech (i.e. when the server distributes text speech to the clients), when shared web pages are called up, when drawing occurs in the Whiteboard Room, and when sounds are played. As all meeting activities are logged through our central server it is easy to graphically reconstruct the “state of the world” at any given time. Again, this is similar to the model described by Ginsberg and Ahuja, in that their view is constructed on the basis of information about when multimedia channels are operative.

In the MultiView, the meeting activity cues are represented graphically and sounds that were played are shown by the command issued to play the sound. Rows are ordered in time sequence; thus row 1 occurred before row 2 and so on. The first column indicates the time at which events occurred. The second column shows the room itself and shows who was present and speaking as well as the time for the participants (note that times shown are based on the geographical location of the original meeting participants). The third column offers a view onto the web document that was being discussed at the time. The fourth column offers a view onto the state of the whiteboard at the time of the discussion. The fifth column shows what sound effects were heard during the represented time “snapshot”. Annotations appear as highlighted comments in the sixth column of the MultiView. Annotations can be added at any time and a time-stamp for these is automatically added as well as the initials of the user that left the annotation. Finally, there is an arrow pointer that monitors where in the meeting the playback onset is currently set.

By selecting an image, the meeting replay can be restarted at the point in time when the events were occurring, and can be reviewed and/or annotated. As the MultiView interface is in a separate web page, it remains visible while the meeting is played back. As events are played, the indicator in the far right column moves through the column to indicate where the playback currently is. To stop playback the user simply clicks on the arrow and playback freezes (as in freeze frame on a video). The arrow is static if no playback is active but moves when playback restarts. The user can drag the arrow to particular rows to continue the playback.

We believe this viewpoint enables viewers to select segments of the meeting to play back on the basis of their own interests. This viewpoint also gives a rich yet easily browsable overview of the meeting activities.

5. Summary

In the above section we described a meeting scenario illustrating how PAVE can support both synchronous and asynchronous activities in a virtual space.

Rather than reading a simple textual summary of what took place in a meeting, interested parties may play back an actual meeting and observe the decision-making process. This playback includes seeing each participant “speak” and change expressions, and watching the surrounding context of documents and whiteboard drawings update as the meeting progresses. Users may insert themselves into the meeting by adding their own contributions, which provides a sense of participation, not just observation. These additions to the meeting are in turn captured, and can be included in subsequent playbacks.

The Replay view, Text view, and MultiView support different goals for navigating through a single meeting, letting the user either participate in, review, or recall meeting events.

Thus far, we have developed only a prototype version of our design to let us experiment with various ways to create and navigate our captured meetings. Implementing a robust version of this system depends on two kinds of infrastructure: a version-control system for storing separate versions of documents and meeting state over time, and a time-line mechanism. Even in our prototype version, preliminary data indicates that the size of storage required to capture a substantial meeting in PAVE is quite a bit less than the comparable space required by video.

A time-line such as TimeWarp's would provide the flexibility to permit timelines to diverge and be rejoined. (Timelines that diverge could arise when distributed groups each revisit a meeting and update it with new interactions simultaneously.) Currently, our simple client-server architecture forces all updates to go through a central server, making it impossible for timelines to diverge. But in a more truly distributed implementation, in which separate groups may update a meeting locally and then periodically synchronize with a remote site, timeline divergence is quite possible.

6. Conclusions

In this paper we have shown how a two-dimensional graphical MOO can be used to support interactions between group members who are spatially and temporally separated. The prototype system, PAVE, captures a complete log of all interactions and shared documents. Given this automatic logging and indexing, and the facility for augmenting captured events, the meeting as a whole is preserved in a way that supports contextually rich meeting reconstruction. The rich hyper-textual log exceeds what can be conveyed through traditional meeting documents like the agenda and minutes. Further, the multiple replay views and the facility for augmentation offers greater flexibility in reviewing and replaying meeting processes than is possible through capturing and replaying meetings on videotape. We argue that this flexibility in how a meeting can be read or revisited offers a strong sense of involvement and participation.

7. Acknowledgements

The authors would like to thank Steve Smoliar for encouraging us to write this paper. We would also like to thank Les Nelson and Eleanor Rieffel for taking part in numerous meetings in our virtual space. We thank Sara Bly and Ellen O'Connor for their encouragement and valuable comments. We thank Polle Zellweger for her helpful comments on an earlier version of this material.

8. References

[1] Bly, Sara A., Harrison, S.R. and Irwin, Susan. Media Spaces: Bringing People Together in a Video, Audio and Computing Environment. Communications of the ACM 36, 1 (January 1993), 28-47.

[2] Carroll, J.M., Mack, R.L. and Kellogg, W.A. Interface Metaphors and User Interface Design. In *Handbook of Human Computer Interaction*, M. Helander (Ed.), Elsevier Science Publishers, 67-85.

[3] Curtis, Pavel. Mudding: Social Phenomena in Text-Based Virtual Realities, in *Proceedings of Directions and Implications of Advanced Computing (DIAC'92) Symposium*, (Berkeley CA, 1992).

[4] Curtis, Pavel and Nichols, David. MUDs Grow Up: Social Virtual Reality in the Real World, in *Proceedings of the Third International Conference on Cyberspace*, (Austin TX, 1993).

[5] Documentum web site: www.documentum.com.

[6] DocuShare home page: www.xerox.com/products/docushare.

[7] Edwards, W. Keith and Mynatt, Elizabeth D. Timewarp: Techniques for Autonomous Collaboration, *Proceedings of CHI 97* (Atlanta, GA, March 1997), ACM Press, 218-225.

[8] Electric Communities web site.: www.communities.com.

[9] The Electronic Meeting Room web page: www.cba.hawaii.edu/emr/home.htm.

[10] Fulk, J. and Collins-Jarvis, L. Wired Meetings: Technological Mediation of Organizational Gatherings. In *New Handbook of Organizational Communication*, F. Jablin & L. Putnam (Eds.), Newbury Park: Sage, 1998.

[11] Ginsberg, A. and Ahuja, S. Automating Environment of Virtual Meeting Room Histories, in *Proceedings of Multimedia '95*, (San Francisco, November 1995), ACM Press, 65-75.

[12] Graham, Stephen and Marvin, Simon. *Telecommunications and the city: electronic spaces, urban places*, Routledge, London.

[13] Harasim, L. *Global Networks: Computers and International Communication*, London and Cambridge, Mass, MIT Press, 1993.

[14] Isaacs, E., Tang, J., Morris, T. Piazza: A Desktop Environment Supporting Impromptu and Planned Interactions, in *Proceedings of the 1996 Conference on Computer Supported Cooperative Work*, (Cambridge MA, Nov 1996), ACM Press, 315-324.

[15] Kendon, A. *Conducting Interaction: Patterns of behavior in focused encounters*, Cambridge University Press, 1990.

[16] Lakoff G. and Johnson, M. *Metaphors We Live By*, The University of Chicago Press, 1980.

[17] Lotus Notes web site: www2.lotus.com.

[18] Manohar, N. and Prakash, A. The Session Capture and Replay Paradigm for Asynchronous Collaboration, Autonomous Collaboration, in *Proceedings of CHI 95*, (Denver CO, May 1995), ACM Press, 218 – 225.

[19] Mark, Gloria, Haake, Jorg and Streitz, Norbert. Hypermedia Structures and the Division of Labor in Meeting Room Collaboration, in *Proceedings of the 1996 Conference on Computer Supported Cooperative Work*, (Cambridge, MA, November 1996), ACM Press, 170-179.