RC 22110 (W0107-010) 6 June 2001 Computer Science

IBM Research Report

Integrating Meeting Capture within a Collaborative Team Environment

Heather Richter, Gregory D. Abowd
Georgia Institute of Technology
801 Atlantic Dr.
Atlanta, GA 30332

Werner Geyer, Shahrokh Daijavad
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Ludwin Fuchs, Steven Poltrock
Boering Mathematics & Computing Technology
P.O. Box 3707
Seattle, WA 98124

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Integrating Meeting Capture within a Collaborative Team Environment

Heather Richter[1], Gregory D. Abowd[1], Werner Geyer[2], Ludwin Fuchs[3], Shahrokh Daijavad[2], and Steven Poltrock[3]

[1]Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332, USA
{hrichter, abowd}@cc.gatech.edu
[2]IBM T.J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, NY 10532, USA
{Werner.Geyer, shahrokh}@us.ibm.com
[3]Boeing Mathematics & Computing Technology, PO Box 3707, Seattle, WA 98124, USA
{Ludwin.Fuchs, Steven.Poltrock}@boeing.com

Abstract. Meeting capture has been a common subject of research in the ubiquitous computing community for the past decade. However, the majority of the research has focused on technologies to support the capture and not enough on the motivation for accessing the captured record and the impact on everyday work practices based on extended authentic use of a working capture and access system. Our long-term research agenda is to build capture services for distributed workgroups that provide appropriate motivation and further understand how access of captured meetings impacts work practices. To do this, we have developed a testbed for meeting capture as part of a larger distributed work system called TeamSpace. In this paper, we discuss the requirements for meeting capture within TeamSpace, describe the initial prototype developed, and report on initial usage.

1 Introduction

Many work practices consist of repeated discussions among teams of people: status is discussed, decisions are made, alternatives are considered, and details are explained. A large amount of this rich, often informal, information that is generated during these discussions often does not get recorded in formal documentation. Yet this information is later useful for providing additional context, details, and decisions surrounding a project. Ubiquitous computing has as one theme the capture, integration, and access of everyday activities, in order to provide a multimedia record of those activities for later perusal [3]. By applying automated capture and access technologies to work discussions, specifically meetings, large amounts of informal project information may be recorded and preserved for later use.

In order to understand how to capture meetings, we must first understand how the captured information might be useful to project team members. Understanding these potential uses is difficult, however. Meetings vary greatly, differing in purpose, formality, and content across domains, organizations and teams. One person may participate in a variety of different kinds of meetings, each with different importance. People have difficulty envisioning how they would take advantage of captured information, and what information would be most useful to their work. Thus, understanding meeting capture will involve putting a system into real use in a variety of situations and domains and being able to adapt the capture services to the needs of a particular project group. Multiple meeting capture prototypes have been built over the past decade, yet little real-world evaluation of these systems has been done. Building a system that can be used in real situations over a long period of time has important design implications. The system must work and provide a valuable service all of the time so we can evaluate under the conditions of authentic experience. The system must be simple and easy to use while supporting many different users. The system will support general purpose meeting activities, but must be evolvable to support more specific activities as the need arises. Additionally, researchers must be able to study how the system is used.

To support a wide variety of meetings, we focus not on supporting specific content, but on general meeting structures and artifacts. To get a better idea of general meeting types and information, we interviewed several meeting facilitators at Boeing who work with teams to improve or enable different types of meetings. We learned that the meetings they deal with are very heterogeneous. Yet, most meetings do have common and simple artifacts, such as agendas, action items, issues, presentations, and minutes. The most important artifacts are often action items, which can serve as valuable minutes of a meeting. Additionally, there was no standard way to create and present most of these artifacts other than word processing software. Thus, users may not be opposed to using new tools if they add benefit.

We have previously argued that for captured information to provide value, it needs to be related to the rest of users' work and everyday tasks [14]. Meeting capture research has primarily focused on supporting and recording meetings. However, users perform many other meeting-related activities that tools can support, and should be able to move easily from meeting-related to other work activities. Additionally, streams of meeting information should become just another form of multimedia information that people will be creating, viewing and sharing. By integrating meeting capture within a larger collaborative work environment that encompasses more than just conducting a meeting, we not only better support the activities surrounding meetings, but also provide a more relevant view of captured information. This additional support will encourage more realistic use of captured information, encouraging users to integrate meeting capture and access as part of their everyday activities.

To that end, we are integrating meeting capture into a larger team environment called TeamSpace, which is being developed in a joint project between IBM Research, Boeing, and Georgia Tech. TeamSpace is a prototype team collaborative workspace for managing shared work processes and maintaining shared artifacts in distributed projects. The overriding goal of the system is to facilitate inter-company teams, which are becoming increasingly common in large projects, such as the development of aerospace systems. Integrating meeting capture into this environment introduces the following additional requirement that we support inter-company distributed meetings.

The goal of this paper is to describe the meeting system of TeamSpace that serves as the foundation for our longer term research agenda to evaluate capture and access in an authentic meeting environment. In Section 2 we discuss related work in automated capture and access. In Section 3 we further discuss the many meeting-related activities we seek to support and the benefits of integration with TeamSpace. In Section 4 we describe the implementation of the TeamSpace prototype. In Section 5 we discuss our experiences in usage thus far and the impact on the prototype. Finally, in Section 6, we conclude with a discussion of our future plans.

2 Related Work

Ubiquitous capture environments have been built in several domains over the past decade: education, personal note-taking, and meetings. In exploring the meeting domain, we are following an approach similar to the eClass project, known formerly as Classroom 2000 [1,2], which studied capture through evolving and evaluating the system through ongoing use in real classrooms. EClass is an education system focused on providing automated access to lecturer's slides and notes, augmented with audio and video. The system has been used to record over 2000 lectures at several universities, resulting in a deeper understanding of why capture works and how it is useful over short and long time periods.

One reason eClass was so successful is that it aided an already important task for students – studying. Motivation for reviewing meeting notes is less clear and more varied, and has been the subject of some study. Whitaker et al. [17] interviewed people on the problems of note-taking during meetings. They found many difficulties that capture could help overcome, such as the failure to note important facts, not enough time to write everything, reduced ability to participate, and the inadequacy of notes for later detailed understanding. Meeting capture could also be used to record information such as design and decision rationale and other informal project information that often disappears over time. Which reasons are most important and will drive the use of a capture system over the short and long term is unclear. We aim to discover this through creating a prototype general enough to enable those motivations to be discovered and then further supported through evolution of the prototype.

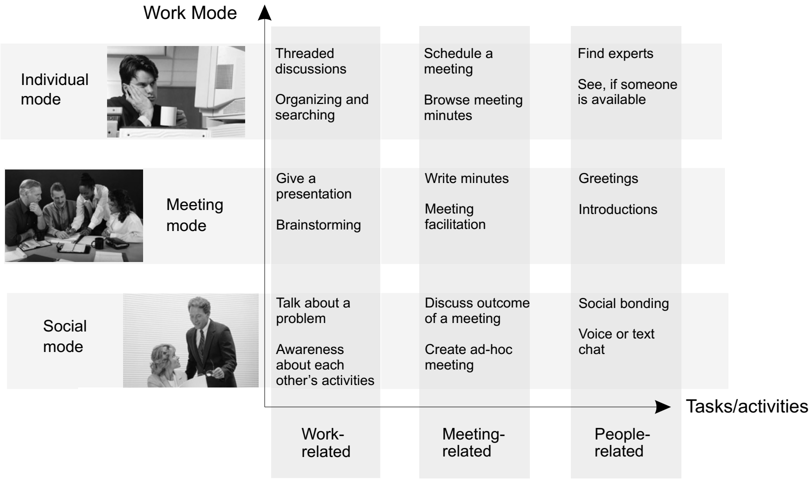
Similar to the education domain, systems such as AutoAuditorium [4] and eSeminar [15] record seminars consisting of a presenter with slides. Reviewing of this captured information is primarily for people who were not in attendance. He *et al.* at Microsoft Research have studied the access of such recorded seminars, including access patterns and auto-summarization techniques [8,9]. Other more general meeting capture systems, such as Notelook [5], Dolphin [16], and Filochat [17] have focused on augmenting presentations or hand-written notes with audio and video streams. Tivoli [12] augmented "domain objects" representing meeting content, with audio. The objects were text and gestures that could be created and manipulated on an electronic whiteboard. However, none of these systems has explored meeting-specific artifacts, such as agendas or action items, nor have they focused on integrating the captured information with other work activities. Additionally, researchers have done little evaluation of these systems in real work environments.

The exception to this is a study by Moran *et al.*[13] with Tivoli, which observed one user preparing reports based on captured meetings. They found that the user did not merely replay meetings, but “salvaged” information by extracting, organizing, and writing information based on multiple records. They further stated that “the development of effective practices of capturing and salvaging meetings must be done by interrelating them with other work practices.” We aim to do this with TeamSpace and plan to evaluate the system with different teams in different environments. It is important to point out that the person salvaging with Tivoli had a real motivation, to write a summary report of a technical topic typically out of his range of expertise. Finding similar motivating aspects for captured meeting review is a challenge nobody has yet faced. Exploring motivations for accessing captured meeting is the long-term research objective that that has driven building our system.

Several commercial distributed conferencing systems, such as NetMeeting [11] and Sametime [10], are beginning to add capture capabilities to conferencing. These systems allow users to conduct distributed meetings by sharing whiteboards, chat, video, and even applications. However, the capture currently involves recording system events and users can then replay the meeting like a movie. This simple form of capture and replay does not support any browsing or searching mechanisms, and thus will be inadequate to help users find pieces of information in even one meeting, let alone a large set of them.

3 Integrating with TeamSpace

The collaborative environment we are integrating with is TeamSpace [7]. TeamSpace is a prototype team collaborative workspace for managing shared work processes and maintaining shared artifacts in distributed projects. The goals of TeamSpace are to support inter-company collaboration through awareness, information sharing, communication, and coordination. TeamSpace aims to support



Work Mode	Work-related	Meeting-related	People-related
Individual mode	Threaded discussions Organizing and searching	Schedule a meeting Browse meeting minutes	Find experts See, if someone is available
Meeting mode	Give a presentation Brainstorming	Write minutes Meeting facilitation	Greetings Introductions
Social mode	Talk about a problem Awareness about each other’s activities	Discuss outcome of a meeting Create ad-hoc meeting	Social bonding Voice or text chat

Figure 1. Matrix of work modes and task.

both synchronous and asynchronous team activities, and to provide a seamless transition between the different work modes and tasks of team members. The classification of the work modes and activities we envision are shown in Figure 1. Team members work in different modes: individual, meeting, and social modes. Additionally, tasks can be work-related, meeting-related, and people-related. Meeting capture systems often focus on supporting some of the work-related activities in the meeting mode through recording notes, audio, and video of a meeting. However, meetings are part of many of the other activities and modes. Individuals asynchronously prepare for meetings, create agendas or presentations, invite participants, or schedule rooms. During a meeting, people greet and introduce each other, take notes, and give presentations; facilitators change the agenda or the flow of discussion. After a meeting, an individual may use the captured material to create minutes or write a report. A user may search through meetings for pieces of information, or people who made certain decisions. Thus, by making meeting capture and access part of a system that supports all of these different kinds of modes and tasks, we can potentially support more meeting-related activities and improve the transition between non-captured and captured activities, and activities that use captured information.

In addition to supporting more meeting-related activities, integrating into TeamSpace will potentially allow us to relate meetings with other team artifacts that are in the environment, such as project schedules, documents, or threaded discussions. Thus, meetings become just another artifact in a large repository of inter-related information. While the current object set in TeamSpace is limited to meeting-related items such as agenda, action items, presentations, meetings, and users, we will be able to add mechanisms for relating meetings with other kinds of artifacts as they are incorporated into TeamSpace.

4 TeamSpace Implementation

In the previous sections we've highlighted how the desire for real-world usage has led us to integrate our capture prototype within a larger work context. We will now demonstrate how we've implemented this prototype as part of TeamSpace. TeamSpace is implemented as a mostly web-based application. This allows it to be accessible from a large number of platforms with no installation. To add meeting capture to TeamSpace, we have added specific meeting-related objects based on the interviews highlighted in Section 1, namely, *Agenda, Action Item, Presentation, Meeting*, and *Person*. We have written additional software to conduct and capture distributed meetings, and replay and review meetings. We have attempted to design all of this software to provide general functionality, yet be flexible and evolvable so that more specific features can be added as we better understand its use by particular project teams. We have also focused on reliability and consistency of the software and interfaces as we want the system to be used and studied over a long period of time. Finally, we wanted to instrument the software to facilitate understanding of users' interactions.

The need to gain as much information for as many users as possible leads us to focus on public (as opposed to personal) meeting capture. While capturing personal meeting notes is certainly important, we would like to make the captured information available to as many people as possible, with as little effort in capturing as possible. One instrumented meeting room can be used my many people for multiple meetings. We also want to require as little instrumentation as possible to enable more locations for capture.

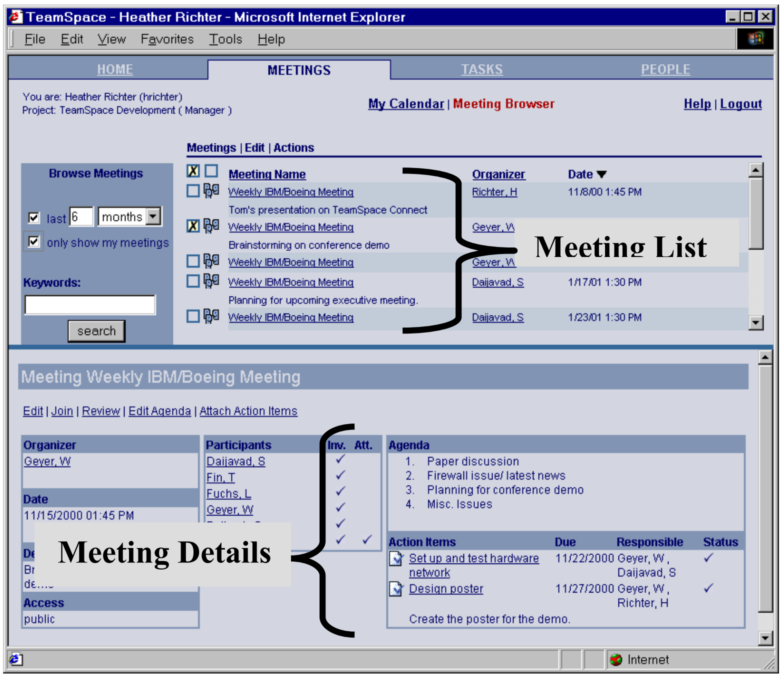
Meeting activities can be thought of in three phases: preparation before the meeting, conducting of the meeting, and later review of the meeting. Each of these phases mainly corresponds to one piece of the TeamSpace prototype implementation. In the following sections we will discuss each in turn. To better illustrate the features of each, we begin each discussion with a running scenario. We then continue the discussion with the user interface implementation, and then with architectural or other implementation issues.

4.1 Meeting Preparation: Main TeamSpace Interface

Bill, the team lead, prepares for a weekly status meeting planned the next day. He checks the meeting information in TeamSpace to make sure the rooms are scheduled and adds a few guest participants. He checks which action items generated discussion last week, and adds those to the meeting. Finally, he adds a small presentation he has prepared for the meeting. The participants are then automatically emailed a meeting invitation.

Meeting preparation is accomplished using the main TeamSpace interface. From this interface users can create, edit, and view any of the objects that TeamSpace supports, such as users, agendas, action items, and meetings. Figure 2 shows a screenshot of this interface. After logging in, users are taken to their starting page which highlights their current day's meetings and their open action items. Users can access additional information using the context tabs of *People*, *Meeting*, and *Task*. The *Meeting* tab provides both a calendar and list view of meetings. The list view is shown in Figure 2, and provides mechanisms to constrain and search the meeting list. Under *Task* a user can view her own action items or browse and search the entire team list. The lower half of the window is a document view for displaying and editing the details of each individual object. In Figure 2, this displays the details for the selected meeting. For the preparation scenario above, Bill would go to the *Meeting* tab to create a new meeting, then enter in all of the information in the document view. Invited participants would then see that meeting on their calendar or meeting list when they logged into TeamSpace that day.

Besides the meeting environment, TeamSpace is intended to support other activities such as project management, document management, and team awareness and communication. In other words, this interface is meant to serve as the main portal for all team activities, including meeting capture and access. However, these capabilities are less mature, and have been left out of the current interface to provide a fully functional yet simplified view of the system. As TeamSpace evolves to include



TeamSpace - Heather Richter - Microsoft Internet Explorer
File Edit View Favorites Tools Help
HOME
MEETINGS
TASKS
PEOPLE
My Calendar | Meeting Browser
Help | Logout
Meetings | Edit | Actions
Browse Meetings
search
Edit | Join | Review | Edit Agenda | Attach Action Items
Meeting List
Meeting Details
Internet

You are: Heather Richter (hrichter)
Project: TeamSpace Development (Manager)

☑ last 6 months
☑ only show my meetings

Keywords:

☒ ☐	Meeting Name	Organizer	Date ▼
☐	Weekly IBM/Boeing Meeting Tom's presentation on TeamSpace Connect	Richter, H	11/8/00 1:45 PM
☒	Weekly IBM/Boeing Meeting Brainstorming on conference demo	Geyer, W	
☐	Weekly IBM/Boeing Meeting	Geyer, W	
☐	Weekly IBM/Boeing Meeting Planning for upcoming executive meeting.	Daijavad, S	1/17/01 1:30 PM
☐	Weekly IBM/Boeing Meeting	Daijavad, S	1/23/01 1:30 PM

Meeting Weekly IBM/Boeing Meeting

Organizer
Geyer, W

Date
11/15/2000 01:45 PM

De
Br
de...

Access
public

Participants	Inv.	Att.
Daijavad, S	✓	
Fin, T	✓	
Fuchs, L	✓	
Geyer, W	✓	
	✓	
	✓	✓

Agenda
1. Paper discussion
2. Firewall issue/ latest news
3. Planning for conference demo
4. Misc. Issues

Action Items	Due	Responsible	Status
Set up and test hardware network	11/22/2000	Geyer, W , Daijavad, S	✓
Design poster Create the poster for the demo.	11/27/2000	Geyer, W , Richter, H	✓

Figure 2. Screenshot of the main TeamSpace interface. The user is viewing the details of one captured meeting. The top is the meeting list view, the bottom pane is the detailed view.

additional artifacts and activities, we will provide mechanisms to relate these to the current set of information.

4.2 Meeting Capture: MeetingClient

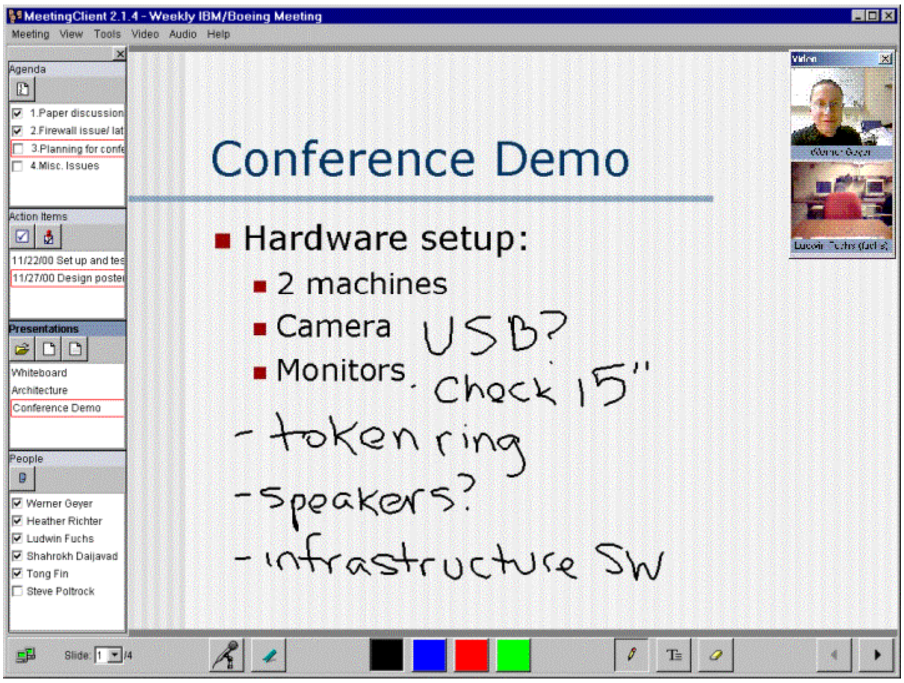
The day of the meeting, team members in Seattle gather in their conference room. Mary, the meeting facilitator arrives a few minutes early to log into TeamSpace and start the meeting. Meanwhile, team members in other locations enter the virtual meeting from their desktop browsers. After team members greet one another and chat for a few minutes, Mary opens the meeting agenda to start. She adds any new items proposed by the team. Pushing the agenda aside, she opens the action item list to get an update on each of the unfinished tasks. As each team member lists their progress, Mary updates the item list, marking off items, changing items, and adding new action items.

The next item on the agenda is a presentation by Bill about an interface problem just discovered between their component and another component. Bill opens the presentation and explains the problem. Working at her desktop in St. Louis, Sally circles a region on one of the components and notes some of the manufacturing constraints that influenced its design. Also from his desktop, Jim draws a sketch to explain the reason for these constraints.

The presentation spawns a brainstorming session for solutions. The team sketches their ideas on the whiteboard, with distributed team members drawing at their desktops.

The meeting capture phase is supported through the MeetingClient interface, shown in Figure 3. MeetingClient is launched automatically on a client's machine from the main TeamSpace web interface when joining a meeting. This client provides viewing, editing, and annotating of agendas and action items, as well as viewing and annotating of PowerPoint presentations. Thus, MeetingClient records events such as joining and leaving a meeting; viewing, editing, and checking off agenda items; viewing, editing, and creating action items, and viewing and annotating presentations. Participants are not required to use or interact with any of these objects. However, the more objects they use, the more events that are recorded, and the more indices that will be created to help in review, as discussed next in Section 4.3.

The panel on the left of Figure 3 provides an overview and navigation of the meeting. The list of agenda items, action items, presentations and invited participants can be seen and individual items selected. The main view shows the selected presentation, or the agenda or action item editor. The toolbar at the bottom of the screen contains the pen and text tools. In the above scenario, Mary would begin the meeting by selecting "Agenda" in the overview panel. She would then edit and rearrange the agenda. Next, she would move to the action item view to go through



Agenda

- ☑ 1.Paper discussion
- ☑ 2.Firewall issue/ lat
- ☐ 3.Planning for confe
- ☐ 4.Misc. Issues

Action Items

- 11/22/00 Set up and tes
- 11/27/00 Design poster

Presentations

- Whiteboard
- Architecture
- Conference Demo

People

- ☑ Werner Geyer
- ☑ Heather Richter
- ☑ Ludwin Fuchs
- ☑ Shahrokh Daijavad
- ☑ Tong Fin
- ☐ Steve Poltrock

Conference Demo

- Hardware setup:
  - 2 machines
  - Camera USB?
  - Monitors. Check 15"

- token ring
- speakers?
- infrastructure SW

Figure 3. Screenshot of MeetingClient. The user is viewing an annotated presentation. The overview bar on the left of the screen shows the agenda, action items, presentations, and participants.

the list of action items. Bill would then view his presentation, which everyone could annotate. Finally, Bill would create a blank presentation to function as a whiteboard for the brainstorming session.

Additionally, MeetingClient provides low-bandwidth video, which is viewed in a separate window, providing real-time awareness of other team members. All of the meeting data and events remain synchronized between clients, and are automatically time-stamped and stored on the server. MeetingClient does not impose any floor control on the distributed users, thus leaving the potential for conflict and unpredictable results. We wanted to keep the interface as simple as possible and will investigate where social protocol is not sufficient and what tool support could help manage the flow of the meeting.

The most important aspect of MeetingClient is to support distributed meetings while intruding upon them as little as possible. Thus, the interface needs to be as simple and easy to use as possible. We've also tried to support common meeting activities without requiring the meeting to be structured in a certain way. Additionally, the interface was designed to work well on both pen interfaces, such as an electronic whiteboard in a meeting room, and on desktop machines. For example, users can add both text and ink annotations to presentations, agenda items, and action items.

In addition to interface challenges, we faced other issues in implementing a capture system for inter-company distributed meetings, such as complicated communication and security. The current architecture of the TeamSpace system is shown in Figure 4. The TeamSpace server consists of servlets and Java Server Pages (JSP) that access and store the data. MeetingClient is a Java application that connects via our own protocol directly to the Conference Server, which is in charge of distributing messages to the clients. The Conference Listener is a passive client that

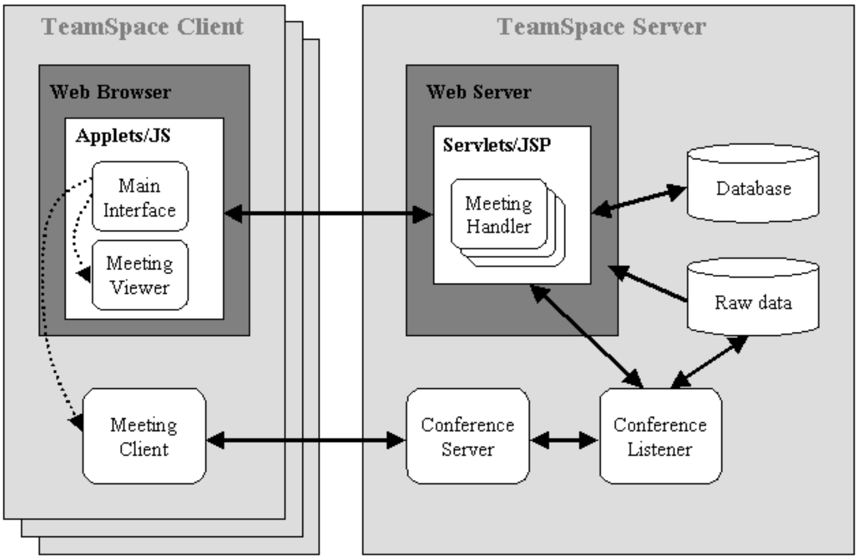


Figure 4. The TeamSpace architecture.

stores a current version of the state so that it can update late-joining clients and store all the data permanently on the server. This solution enables the meeting clients to communicate through firewalls given that the server is located on the open Internet and the firewall allows TCP and HTTP traffic. It also supports and synchronizes any number of clients in one meeting. We further discuss security and communication issues of this architecture in Section 5. While we use our system to distribute video images, to eliminate transmission delays we are not transmitting and mixing audio ourselves. We expect participants to use a conference call that is then input to any one of the meeting clients so the audio can be recorded and digitized. The Conference Listener stores the audio, video, and event streams as raw data on the server's file system. All other information, such as action items and meeting descriptions, are stored in a database.

4.3 Meeting Access: MeetingViewer

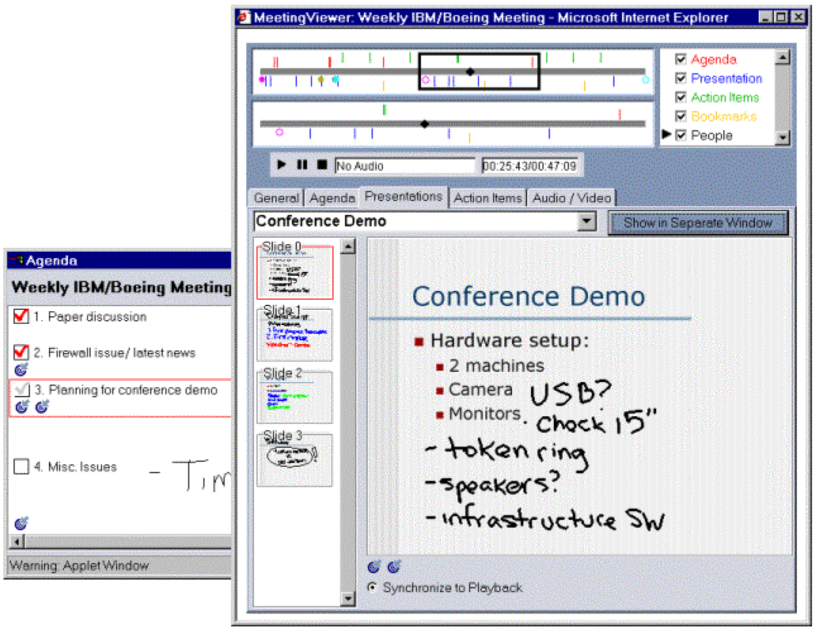
During the meeting, the team was split between two different solutions to their problem and needed more discussion to make a decision. Pat and Jim meet later that day to further discuss one of the proposed solutions. During their meeting, which they also capture, they recall a contact Sally mentioned during the earlier meeting. They open up the meeting and browse through it by using a timeline that indicates the spots where Sally was talking. They easily find and replay the portion where Sally was sketching on the components and hear her mention her co-worker Dave. The group looks up Dave's contact information and gives him a call.

Chris had to leave the weekly status meeting early. Before he tackles his new tasks, he returns to the meeting records to listen to the portions he missed. He skims the meeting with a time slider by jumping from one agenda item to the next. Then he dives into Bill's presentation, using a thumbnail navigation, to replay the portion of the presentation where the group talked about the manufacturing constraints. Chris also listens to the comments Dave made to Pat and Jim during their conversation.

One year later, Bill is leading a team that runs into a similar component interface problem. He asks Harry, one of his team members, to look at the problem the old project had, and why they chose their solution. Harry reads the documentation, accesses the meeting records and replays pieces of various discussions, and prepares a presentation using some of the older material to present his findings.

After a meeting is completed, the meeting records are automatically available for retrieval. In this prototype we have focused on retrieving meeting details of one or several meetings. Users can select completed meetings in TeamSpace and launch a MeetingViewer applet to view and playback these meetings.

The MeetingViewer, shown in Figure 5, integrates all of the meeting information based on time. The viewer uses a two-scale timeline for navigating a set of selected meetings, providing random access playback. The timeline is painted with interesting events as both a visual summary of the meeting, and as an aid for navigation. Interesting events currently are people joining and leaving, agenda items being discussed, action items visited or created, and slides visited, but could include any future events such as people speaking and keyword locations. Users can control



Weekly IBM/Boeing Meeting

☑ 1. Paper discussion

☑ 2. Firewall issue/ latest news

☑ 3. Planning for conference demo

□ 4. Misc. Issues – Tim

Conference Demo

Conference Demo

- Hardware setup:
  - 2 machines
  - Camera USB?
  - Monitors. Check 15"
- token ring
- speakers?
- infrastructure SW

Figure 5. Screenshot of MeetingViewer showing a single meeting. MeetingViewer can also be used to view multiple meetings.

which of these events they view and can use the events to find relevant portions within a meeting to playback. Playback of a meeting not only involves playing the audio and video, but also involves playback of all of the recorded events of a meeting such as slide visits or agenda item discussion.

The remainder of the meeting information is displayed on a series of tabbed panes for each of the objects related to the meeting, including descriptions and summaries of the meeting, agenda, presentations, action items, and video images. These panes are a very general approach for displaying a large amount of related information. However, to enable customized views, each pane can be opened in a separate window, moved and resized. In this way, users can view any subset of the information they wish at once. Additionally, as we add more objects to TeamSpace, we can easily add more meeting-related objects to this interface as another tabbed pane, such as documents that were reviewed or referenced during the meeting.

When reviewing meetings in the short term, users can potentially use any kind of context to find a piece of information – from a note, to when someone spoke, to what the general subject matter was. For this reason, we started with a very general review interface. We anticipate that as we learn more about the types of information users need for various tasks, we can design task-oriented views that are simpler and more integrated. While detailed context is appropriate to aid short-term browsing and search, longer-term searches, such as Harry's in the above scenario, will probably require different mechanisms that emphasize summaries. We would like to investigate additional visualizations to navigate and search a large set of meetings as our set of captured material grows.

4.4 Summary

While we have not implemented every feature our scenarios describe, we feel we have a reasonable prototype that successfully allows users to do a number of meeting-related activities. The capture system supports capturing both work-related artifacts, such as presentations, but also meeting-related artifacts, such as agendas. The main interface serves as the starting point for many of the individual mode activities, such as preparing for a meeting and reviewing meeting information. The various objects are inter-related and can be viewed in multiple ways, but the interface could still be improved to make this more explicit. For example, when the user is viewing an action item, she should be able to view where that item was discussed in a meeting and jump directly to replaying that meeting. Additionally, we would like to extend search to searching other meeting content, such as slide texts. Evaluation of the system will hopefully help prioritize the features that can provide the most benefit to users. Finally, as TeamSpace evolves to add additional objects and activities, we can provide mechanisms to relate these to meetings and also integrate them into the capture and access software without changing the structure of their interfaces.

5 Initial Experience

Our research team has been using TeamSpace to conduct and capture our distributed meetings for the past six months. We have successfully recorded over a dozen weekly meetings, as well as a distributed presentation for a yearly project review, and two 4-hour strategy sessions. Typically the meetings were held at three separate locations, in both meeting rooms and at desktop machines. The meeting environment has been useful for brainstorming and for discussing presentations together and team members have revisited several meetings to view the annotations we made, or review a missed meeting.

The first thing to examine is how TeamSpace has affected our meetings. Prior to the completion of the prototype, we did not make much use of an agenda, although team members would often privately note items to discuss prior to the meeting. These items are now publicly listed in the agenda, yet this hasn't been too important as most meetings only contain a few agenda items. However, the agenda was used extensively and much appreciated for the 4-hour strategy meetings. Additionally, we now create information to be shared during a meeting in PowerPoint®. As we did not have a standard way to share information previously, this has not been a problem, although we would certainly like ways to share information in other formats. Finally, we are experiencing an interesting phenomenon common to distributed collaborative systems – knowing what the other party is seeing. One frequently asks if others can see that he has joined or made an annotation. While this was often necessary when the prototype was not stable, we continue to ask these questions even now. We need to investigate how the system can give users these cues so that they do not have to interrupt the meeting to ensure that the system is still working.

Captured information has been most useful for team members who missed the meeting. We also find that we do have moderate needs to review information in our meetings. Note-taking prior to TeamSpace was sparse. Yet, the annotated slides have been useful to review brainstorming or edit a presentation. We do occasionally discuss an action item and review the list the following week to make sure we didn't forget to do anything. Thus, while a group such as ours does not do large amounts of meeting review, we do rely on the system to record our activities and find TeamSpace useful. One team member even likes to use TeamSpace to record other non-distributed meetings. Capture seems to add a comfort level of knowing that the meeting is being recorded even though we might not know if we will review it.

Our usage has highlighted several deficiencies in our interface designs, and has led to revisions of both the main interface and MeetingClient. In a meeting, navigation, particularly navigating between different pieces of information, such as an agenda item and a slide, was awkward. This navigation needs to be extremely simple, particularly for those using a pen on an electronic whiteboard. We will not learn much about capturing meeting artifacts if information such as agendas or action items is not captured because it is too difficult to use during a meeting. For example, we found that it would be useful for the agenda to be more easily viewed so that moving on to the next agenda item does not require multiple user actions. To address these difficulties, we added the overview panel to the capture client - the left hand panel in Figure 3. This enables a quick overview of meeting information and easy switching between viewing any items in more detail. The look of the main TeamSpace interface was simplified by adding multiple ways to view items, such as the calendar and list views for meetings. In this way, only relevant information for each user is shown and can be modified to reduce information overload.

We have also encountered other implementation problems in building such a distributed system. Enabling inter-company collaboration is more difficult than we expected [6]. Security is a major issue in any system, and is only compounded by cross-firewall communication. Security infrastructures are in place to inhibit the flow of information outside of the company, yet collaboration requires circumventing security mechanisms without violating their principles. We first attempted to use existing meeting conferencing software to handle communication and distribution. However, this did not work across companies and we eventually created our own communication protocols. While our current prototype can communicate through firewalls, the information is not secure enough for real use as the server resides on the open Internet. We are exploring alternative architectures to address security concerns. We are also continuing to investigate if existing conferencing software can meet our communication requirements so that we may take advantage of other existing features.

6 Conclusions and Future Directions

Our long-term research agenda is to discover the impact of automated meeting capture for distributed workgroups. While meeting capture has been a popular theme in the ubiquitous computing research community, there has been relatively little work reporting on the use of a capture system for an extended period of time by a distributed workgroup. There are several requirements that must be met by a research testbed intending to explore authentic meeting capture in this kind of setting, including evolvability of a general purpose meeting support system that is tailorable to the work practices of a specific team, the distribution of capture over several work sites, the integration of meeting capture within the larger electronic workgroup support system, and a reliable system that can be used over a long period of time to permit observation by a research team. To meet these requirements, we have built a meeting capture system as part of the TeamSpace project. In this paper, we described the features of this meeting capture system, demonstrating why it is an appropriate vehicle for further exploration in this domain.

While our experiences with TeamSpace have highlighted several technical and usage issues, we need much more extensive usage to understand the impact of such a capture system. TeamSpace now provides us a research vehicle for gaining that experience. We will be deploying the current prototype to multiple teams in Boeing, IBM, and elsewhere. We have found that our implementation has made instrumenting the various pieces of software easy using a servlet that will store log information that the software sends to it. We now need to understand the kinds of analysis we can do with this data.

As we better understand users' motivations for capturing and reviewing meetings, we can evolve TeamSpace to better support these motivations. There are already a myriad of features and capabilities we could add. We have not yet implemented a visualization for browsing a large set of meetings. We could integrate with other technologies, such as speaker identification or auto summarization, to make access more powerful. It is difficult to prioritize which of these features will provide the largest benefit to users, and thus help highlight important uses and motivations.

Designing and implementing a research prototype for real-world evaluation of meeting capture has been challenging. Our research goals led us to build a system that integrates the meeting capture into a larger team workspace, which we hope will encourage realistic and repeated usage. We have implemented a general solution that we feel can be used and evolved over time. We continue to face issues of any system that is deployed for real use, such as usability, security, and scaling. But building such a system becomes increasingly important in ubiquitous computing as we seek to understand its impact in people's everyday environments.

Acknowledgements

We would like to acknowledge the other members who have contributed to the TeamSpace project: Carolyn Brodie, Tong Fin, and Tom Frauenhofer at IBM T.J. Watson Research and Khai Truong at Georgia Tech. This work was partially supported by DARPA/ITO under the Information Technology Expeditions, Ubiquitous Computing, Quorum, and PCES programs, and by the National Science Foundation (grants 9703384, 9818305 and 0070345).

References

1. Abowd, G., Atkeson, C., Brotherton, J., Enqvist, P., and LeMon, J. "Investigating the capture, integration, and access problem of ubiquitous computing in an educational setting," in Proc. CHI 98, 1998.
2. Abowd, Gregory D. "Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment," *IBM Systems Journal*, Special issue on Pervasive Computing, **38**(4): 508-530, October 1999.
3. Abowd, Gregory D., Elizabeth D. Mynatt. "Charting Past, Present and Future Research in Ubiquitous Computing," in *ACM Transactions on Computer-Human Interaction*, Special issue on HCI in the new Millenium, **7**(1): 29-58, March 2001.
4. AutoAuditorium Homepage: http://www.autoauditorium.com.
5. Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L. "Notelook: Taking Notes in Meetings with Digital Video and Ink," in Proceedings of ACM Multimedia '99, 1999.
6. Fuchs, Ludwin, Werner Geyer, Heather Richter, Steven Poltrock, Tom Frauenhofer, and Shahrokh Daijavad. "Enabling Inter-Company Team Collaboration," in Proceedings of the 10th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WetICE 2001), June 2001.
7. Geyer, Werner, Heather Richter, Ludwin Fuchs, Tom Frauenhofer, Shahrokh Daijavad, and Steven Poltrock. "A Team Environment Supporting Capture and Access of Virtual Meetings." In Proceedings of Conference on Supporting Group Work (Group 2001), September 2001.
8. He, L., Sanocki, E., Gupta, A., and Grudin, J. "Auto-Summarization of Audio-Video Presentations," in Proceedings of ACM Multimedia '99, 1999.
9. He, L., Sanocki, E., Gupta, A., and Grudin, J. "Comparing Presentation Summaries: Slides vs. Reading Vs. Listening," in Proceedings of CHI 2000, April 2000.
10. Lotus Sametime Homepage, "Real-Time Collaboration That's Fit for Business", URL: http://www.lotus.com/Sametime, November 2000.
11. Microsoft NetMeeting Homepage: http://www.microsoft.com/windows/netmeeting.
12. Minneman, S., Harrison, S., Janssen, B., Kurtenbach, G., Moran, T., Smith, I., and van Melle, B. "A Confederation of Tools for Capturing and Accessing Collaborative Activity," in Proceedings of ACM Multimedia '95, 1995.
13. Moran, T.P., Palen, L., Harrison, S., Chiu, P., Kimber, D., Minneman, S., vanMelle, W., and Zelweger, P. "'I'll Get That Off the Audio:' A Case Study of Salvaging Multimedia Meeting Records," in Proc CHI '97, Atlanta, GA, 1997.
14. Richter, Heather A., and Gregory D. Abowd. "Automating the capture of design knowledge: a preliminary study." Technical Report GVU-99-45, Georgia Institute of Technology. December 1999.
15. Steinmetz, Arnd, and Martin Kiezle. "The e-Seminar Lecture Recording and Distribution System," in Proceedings of SPIE (International Society for Optical Engineering) Vol. 4312, Multimedia Computing and Networking, San Jose, CA 2001.
16. Streitz, M.A., Geibler, J., Haake, J.M., and Hol, J. "DOLPHIN: Integrated Meeting Support across Local and Remote Desktop Environments and LiveBoards," in Proceedings of Computer Supported Collaborative Work Conference (CSCW '94), 1994.
17. Whittaker, Steve, Patrick Hyland, and Myrtle Wiley. "Filochat: Handwritten Notes Provide Access to Recorded Conversations," in Proceedings of CHI 1994, Boston, MA, 1994.