DESIGNING AND EVALUATING MEETING CAPTURE AND ACCESS SERVICES

A Dissertation
Presented to
The Academic Faculty

by

Heather Richter

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
April 2005

DESIGNING AND EVALUATING MEETING CAPTURE AND ACCESS SERVICES

Approved by:

Dr. Gregory D. Abowd, Advisor
College of Computing
Georgia Institute of Technology

Dr. Mark Guzdial
College of Computing
Georgia Institute of Technology

Dr. Elizabeth Mynatt
College of Computing
Georgia Institute of Technology

Dr. Colin Potts
College of Computing
Georgia Institute of Technology

Dr. Jonathan Grudin
Microsoft Research

Date Approved: April 13, 2005

ACKNOWLEDGEMENTS

I would first like to thank my dissertation committee for all of their time and suggestions. I greatly appreciate their insight and comments that have influenced and improved my research.

I am so fortunate to be a part of the Ubicomp research group. I do not know what I would do without my fellow labmates to answer all of my questions, give me feedback, listen to me when things are rough, and entertain me when I have been working too long. Jason, Jen, and Anind, Jay, Gillian, Giovanni, Kris, Lonnie, Xuehai, Shwetak, Julie, and especially Khai Truong, I will miss you all. I hope that as our academic family grows and spreads out, we can continue to help and support each other, and more than anything, stay friends and colleagues.

I also want to thank all of my other officemates and friends at Georgia Tech, especially Idris Hsi and the rest of the INCITE group, who have supported me and given me a great place to work. They have shown me that getting a Ph.D. is more than just the research, it's the relationship we build up along the way.

Finally, and most importantly, I want to thank my advisor, Gregory Abowd. He has given me the freedom to explore the things I wanted to explore, and supported me in everything that I have done. But even more than that, he has been a wonderful mentor, both with advice and by example. The enthusiasm he brings to his work and his students has infected us all. I hope that I can inspire my students the way he has inspired me. I look forward to his continued inspiration as I move into the next phase of academic life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMMARY

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, but 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. One of the themes in ubiquitous computing is the capture, integration, and access of everyday activities. By applying automated capture techniques to work activities, we can potentially record a large amount of information for later use, some of which may not otherwise get recorded. Meetings are an excellent activity to capture, as they are a crucial communication and coordination activity of teams, and a ubiquitous activity in the workplace.

In order to understand how to capture meetings, we must first understand how the captured information might be useful to project team members. However, understanding these potential uses is difficult. Meetings vary greatly, differing in purpose, formality, and content across domains, organizations and teams. 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.

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 but 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. In this thesis, we are designing and implementing capture and access prototypes for several domains. We are evaluating these prototypes in realistic use to gain an understanding of the motivations for using these prototypes, the patterns of reviewing meetings, and the effect of capture on the meeting.

CHAPTER I

INTRODUCTION AND MOTIVATION

People in the workplace generate a large amount of information. For example, in systems development, teams gather requirements and create design specifications. Managers generate schedules and milestones. Quality control groups create and execute test plans. Much of this information is documented in structured, standard ways. However, much information can be left out of these documents. Teams discuss status, brainstorm and sketch possibilities, consider alternatives and make decisions based on various assumptions along the way. Tasks are distributed and completed by many different members of the team and can, in turn, affect future tasks. Documenting all of this rich, often informal information is cumbersome. Yet this information may later be valuable to provide additional context, details, and decisions.

Much of this informal information is generated during the natural activities that occur in the workplace. Many of these activities involve teamwork, groups of people discussing and sharing artifacts. One of the themes in ubiquitous computing is the capture, integration, and access of everyday activities (Abowd, 2000). The purpose of recording this rich multimedia information is to free humans to do what they do best — attend to, synthesize, and understand what is happening around them, all with full confidence that the specific details will be available for later perusal. By applying automated capture techniques to work activities, we can potentially record a large amount of information for later use, some of which may not otherwise get recorded. Meetings are an excellent activity to capture, as they are a crucial communication and coordination activity of teams, and a ubiquitous activity in the workplace.

Capturing meetings has several potential uses. In the short term, automated capture can give people a better memory of their meetings, potentially leading to more accurate work performance. In the long term, the large amount of captured material provides a rich, informal record of decisions, details, and events of the life of a project. In order to understand how to capture meetings, we must first understand how the captured information might be useful to project team members. However, understanding these potential uses is difficult. 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 differing degrees of 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 capture and access services into realistic use in a variety of situations and domains and examining the patterns of behavior that evolve as those services are put into to use. We are particularly interested in the domain of software engineering, a domain rich with complex knowledge where different discussions occur throughout the lifecycle and affect the software under development.

1.1 Meeting Recording

The purpose of meetings is to move group activities forward towards a common goal. Meetings help team members to coordinate their work, come to a shared understanding of their work, and focus on their task (Minneman, 1995). Team members present information to others and collaborate with each other through reviewing, evaluating, discussing, problem solving, and deciding. There are also social reasons to meet, such as the need to belong, to achieve, and to make an impact, or the desire to communicate, build, and share a common reality.

The rich amount of information contained in meetings suggests that there is some value in documenting meeting activities and their outcomes. Private note taking is common practice in meetings, and at times a designated person prepares meeting minutes, which are usually a narrative meeting record. Good meeting practices, which are mainly based on heuristics, emphasize the importance of preparing meeting records and suggest capturing at least all the decisions made, all the action items assigned, and all the open issues. As such, meeting records help create a shared group memory and make meetings more efficient by:

- decreasing the need to revisit decisions made,
- easing recall of open issues and deferred items,
- and providing increased confidence that action items will be done.

Personal notes of a meeting serve as a memory aid for individuals to remember important facts, actions, ideas, and decisions but are generally not useful for persons other than the author. In contrast, meeting minutes are a narrative public record of participants, decisions and issues. Complete minutes require a dedicated recorder with limited ability to actually participate in the meeting. Additionally, minutes do not capture individual information needs and are subject to the interest, contextual knowledge, and summarization ability of the recorder. Collaboratively prepared artifacts also serve as outcomes of meetings, such as sketched diagrams or brainstorming. Yet often the process of how and why those artifacts were prepared is lost.

Technology can potentially overcome the difficulties with traditional meeting recording, reducing the burden on the meeting participants. However, we are still a long ways from having machines that can produce as concise or readable a summary as the minutes produced by a human recorder. Thus, automated recording of meetings will involve media other than just text minutes, and will require particular interaction methods for reviewing such recordings.

Automated audio and video recording is a common choice for recording meetings. If done properly, audio and video together can provide a comprehensive meeting record that allows people to see who was present and what was discussed. This recording technology is unobtrusive and generally does not require any further interaction during the meeting once recording has been started. However, even digital audio/video recordings often resemble the functionality of a VCR tape, which requires people to either watch or skim the entire recording to find information. In other words, there is no structure other than time to help navigate through the meeting record to find detailed information. This may be acceptable for people who were not at a meeting, and thus want to review the entire event. Yet this is time-consuming if one instead wants to focus on specific topics or details of the meeting.

Meetings are also often supported by additional technology, such as electronic whiteboards in conference rooms, or application sharing for distributed meetings. When recorded, these additional media are either presented statically, such as the image of the whiteboard, or presented in simple VCR-like playback. Thus, while more information is being recorded, that record is still difficult to review. Multimedia records of meetings will only be generally useful if tools or technologies exist that help users avoid replaying much of what has been recorded (Wolf, 1992). In other words, the utility of the record must not be outweighed by the effort of reviewing it. People need to be able to quickly browse and access those elements of the record that are of interest to them. This kind of directed browsing, searching, and visualization of meeting records requires meaningful indices that act as semantic pointers into the multimedia record.

1.2 Meeting Indices

An index is essentially a time-point that meaningfully (in theory) describes or signals a point in a recording. An index can aid in navigation by providing a random-access point to which someone can jump directly. Minneman et al. (Minneman, 1995) classify indices for captured material into four broad classes: *intentional annotations*, *side-effect indices*, *derived indices*, and *post-hoc indices*. *Intentional* indices are indices that participants create during an activity for the purpose of marking particular time points in that activity, such as sequential note taking where the notes are time-stamped. *Side-effect* indices are indices that are created through activities whose primary purpose is not indexing, such as recording slide changes in presentation software. *Derived* indices are generated through signal analysis of the records, such as speaker identification or cut detection. Finally, *post-hoc* indices are intentional indices produced through later user access to the meeting record.

We wish to modify this classification to cover additional possibilities (Geyer, 2003). Indices can be created at different times — *online*, occurring during the synchronous activity, and *offline*, occurring after the activity. The intent of the user can be difficult to determine during a meeting; for example, the user may not be fully aware that the notes they are taking are being time-stamped to serve as an index. Thus, we do not classify indices based on user intent. Instead, we further classify indices based on the underlying technology creating and gathering them. *Explicit* indices are those created through user actions with the capture and access services, whether or not users intend or realize their actions are creating a time-stamped event. One example is when the user takes notes or draws on the electronic whiteboard provided by the meeting software. *Derived* indices can be considered more implicit, sensed or inferred through other means and are not related to direct user actions, such as using sensors to detect who is in the meeting room or video analysis to detect slide changes. A summary of this classification is shown in Figure 1.1.

	Explicit	Derived
Online	e.g. handwritten notes, joining and leaving	e.g. slide detection, individual location
Offline	e.g. posthoc bookmarking	e.g. speaker identification, keyword spotting

Figure 1.1. Indexing classification with examples.

Many of the approaches to indexing meeting records, such as the Meeting Room project (Schultz, 2001), have focused on derived indices such as speaker identification, transcription, and slide or scene change detection. This approach is valid and useful to improve the accessibility of a meeting record, but is also technically challenging and error prone. Moreover, while the meeting record is better indexed, allowing for easier browsing, the record does not directly relate back to the work people are doing. Thus, the meeting recording will still exist in isolation from the rest of the team's activities, reducing the chances of meaningful review. To better relate to the work that a team is doing, the captured record must be linked, or indexed, by the specific work artifacts and activities of the team members. Creating many of these links using derived approaches will be very challenging. Instead, we are focusing on explicit indexing: recording users' interactions with their specific artifacts during the meeting. Through this, we can allow artifacts to be created and manipulated during a meeting, and in turn, act as indices into the meeting afterwards. The resulting recording is thus structured in more semantically meaningful ways, providing additional means for understanding the outcomes of a meeting or for locating specific information in the recording. The challenge is in capturing this structure through supporting natural activities with meaningful artifacts when possible, rather than imposing structure or artifacts on a group.

1.3 Results and Contributions

This thesis focuses on evaluating several capture and access prototypes in realistic settings. In order to do that, we must first design and build useful enough systems. As we have previously argued, a useful meeting recording needs to be related to the context of work – the activities and artifacts of the group. To do this we are supporting the specific activities and artifacts in the meeting, explicitly capturing the interactions by the users, and thus creating meaningful indices into the recording. The indices we capture and use for review will obviously influence the usefulness of the system, and will be an underlying aspect of any of the evaluations we perform.

We will design, build and use capture and access prototypes in three different domains, and evaluate various aspects of their use. Introducing technology into a meeting will change certain aspects of that meeting. First, we attempt to support already existing meeting practices through software in order to capture the specific activities and interactions that occur. While the goal is to make this support as seamless and natural as possible, the software will still be different from the group's existing practices. In each domain, we explore a different set of artifacts and activities, demonstrating the range of activities we can support with technology, and how the captured indices structure the meeting records. The evaluations of these prototypes attempt to address two broad questions:

How do users browse, search, and replay captured meeting content? What information should be captured needs to be informed by what information is used to access meeting content. Thus, we need to understand how people can navigate meeting records and find information. What information do users view? How do they navigate the recording? Much of our experience with multimedia is with VCR-like playback or web-like navigation, yet this is cumbersome for quick review. How can users more efficiently find specific meeting details? What context are they using to recall details? What indices are they using? What do they gain from the audio and video vs. the more static information such as slides and notes? What are the patterns of replaying the multimedia? How does this use evolve over time and experience?

What are the motivations for reviewing captured material? The most important aspect in building any computing services is to understand the real needs of the users. The difficulty in ubiquitous meeting capture and access is that thus far those needs are based on intuitions or self-reported desires of users. We have no understanding of the real utility of capture and access services and the impact on users. What tasks are users trying to accomplish with captured meeting information? What value will users really find in captured meetings? Is the utility worth the effort of reviewing? A challenge is that any one meeting alone is not going to be useful enough to a group to show benefits.

Instead, capture and access becomes useful only when expected, depended upon, and adopted into everyday use. Thus, we also first need to understand what capture and access features lead to repeated usage of these services as well as what discourages use in order to begin to understand overall motivations.

Finally, meeting capture can have a large social impact on participants. Knowing that their words are being recorded may have many complicated effects. Meeting participants may be uncomfortable with the idea of being recorded. Participants may take more care in what they say, or may simply opt to speak less. Group members may feel less inclined to chat or gossip or talk informally as those comments would also be recorded. Those who were not at the meeting would now have the opportunity to hear what they missed, but may still misinterpret comments or take them out of context. These feelings will likely be varied and personal. As we explore capture prototypes in real use, we need to begin to understand what effects these can really have on a meeting and how that would affect the use of capture services.

1.4 Purpose of Research and Thesis Statement

Ubiquitous capture environments have been researched and built for the past decade in the meeting and education domains. However, little work has been done in understanding how captured meetings impact work practices. In this thesis we aim to advance this understanding by prototyping several meeting capture and access systems for different types of meetings and evaluating them in realistic environments. The evaluations will focus on several different aspects of capture and access, namely, the capture of domain-specific indices, the access patterns, and the overall use of the captured information.

Our thesis statement is as follows:

*The **natural activities of a meeting** can be captured as a series of events to later serve as **indices** into the meeting recording. The use of these indices will be effective for **searching, navigating, and replaying** the captured meeting to aid **various work tasks**.*

As stated earlier, we are focusing on explicit indices to index the captured meeting record. In each of the domains we explore, we build prototypes that both record the meeting through a variety of media such as audio, video, and ink, as well as support the natural activities and artifacts of that meeting. As users perform those activities and interact with artifacts through the software, we capture these interactions as discrete events. These time-stamped events then later serve as access points into the record, and as a link between the captured meeting and the artifact. Through building capture and access services in this way, we aim to provide prototypes that can be used and evaluated in realistic settings.

1.5 Overview of Thesis

Meetings vary greatly in domain, artifacts, and activities involved. Meetings can be fairly broad and unstructured, involving status updates, information sharing, brainstorming and decision making. Other meetings follow specific structures and use domain-specific artifacts. Understanding meeting capture and access should involve examining these services in a variety of domains. In this thesis, we build and evaluate meeting capture and access prototypes for three different types of meetings.

SAAM: We begin with capture of software architectural analysis meetings following the Software Architectural Analysis Method (SAAM). This work was begun with an interest in capturing the design rationale surrounding software architectures. We built a capture prototype supporting the creation and manipulation of architectural diagrams during structured analysis meetings. Additionally, we built a prototype access interface around video recordings of several real analysis meetings and a summary document of that meeting. With these prototypes, we examine the use of a specific artifact, namely the software architecture, as a captured artifact and as an index into the recorded meeting. We also demonstrate how such records can be navigated and useful beyond standard documentation. These early results were encouraging. However, this is a relatively rare meeting for a design team, limiting our ability to gain real, long term use.

TeamSpace: In response to the difficulties in performing long term evaluations in the SAAM exploration, we focused on supporting a wider variety of everyday discussions. Thus, in this exploration, we describe work in the capture and access of general meetings along with longer-term evaluations. The system, TeamSpace, is a collaborative workspace for distributed teams, with integrated capture and access of meetings. Supporting general meetings involves supporting general meeting artifacts, namely participants, agendas, action items, and documents. TeamSpace provided us an opportunity to put capture and access services into everyday use to examine the behaviors that evolve and the motivations for its use. Additionally, we performed a detailed study of the behavioral patterns and interface support in the TeamSpace access application, MeetingViewer. Through these evaluations, we found that many of these everyday discussions are low-need situations, where users have infrequent needs for accessing meeting content.

Tagger: Finally, we returned to the software engineering domain to examine meetings with higher needs for document the details of the discussion. We explore capturing requirements gathering meetings for large development projects, called knowledge acquisition sessions. Like SAAM, this type of meeting is also a fairly structured and specific meeting, with specific expected outcomes such as domain models and requirements specifications. However, in this case, the artifacts do not yet exist. Thus, the indices we gather are much more user-driven – users assert software-related concepts to index portions of discussion such as terminology, objects, and issues. We examined the feasibility of Tagger, a prototype to capture knowledge acquisition sessions, automatically transcribe the audio of the discussion, and provide the transcription to meeting participants to tag. We also explored the benefits of captured sessions, comparing the use of TagViewer, an access prototype, with the use of plain video, in creating a requirements document.

In the remainder of this thesis, we begin in Chapter 2 by summarizing the background and related work in meeting capture, ubiquitous capture and access, and evaluations of existing systems. In Chapters 3, 4, and 5, we discuss each of the three case studies in detail, including the evaluations and results of each. Finally, we conclude in Chapter 6 with a general discussion of the important issues and contributions raised in each study, and the design guidelines we have learned throughout.

CHAPTER II

BACKGROUND AND RELATED WORK

In this chapter, we discuss some background and work related to meetings and capture and access. We begin with research in meetings and their memory outside of ubiquitous computing, specifically note-taking and design rationale methods for recording the details of work discussions. This research motivates the potential benefits of automated meeting capture and access. We then define and discuss ubiquitous capture and access, detailing many meeting and other capture systems. Finally, we discuss what we have learned from previous studies of capture and access systems in use.

2.1 Meetings and Their Memory

Meetings are a frequent and important activity in the workplace. They often contain important information regarding issues and decisions, tasks to be accomplished, and information that needs to be shared. In order to remember meeting information, people often create external records, but often just rely on their memories and use memory cues to facilitate the recall of information. As part of their work on multimedia meeting retrieval, Jaimes et al. surveyed what types of items people currently use to review meeting contents as well what people remember about meetings (Jaimes, 2004). They found that documents, memos, minutes, and just asking someone were frequently used, and that video is currently rare. They also categorized the reasons for using video to record meetings as Verify, Understand, Reexamine, Keep record, and Recall. Finally, to examine the memory cues people actually do remember, they asked subjects a number of questions regarding a recent meeting. They found that certain items, such as the location of the meeting room, position and names of meeting participants, and major topic names were easily remembered. Subjects also had good memory of "impressive" slides, even 5 weeks later, and had better memory of activities they performed themselves. Memory of information such as the date of the meeting, starting and closing times, room number, and details of the dress, expressions or gestures of the participants was relatively poor. Knowledge of these cues could inform the design of meeting access systems and the kinds of indices that are recorded.

People have always made records of meetings, most commonly by taking personal notes. We continue to discuss meeting records with studies of note-taking and its problems. Additionally, the knowledge management community has started looking at meetings as an important source of knowledge within the workplace and has proposed methods for capturing this knowledge. Some of these methods are based on design rationale capture methods that aim to document issues and decisions as they are discussed. While we will not be able to compare these methods directly to capture and access in this thesis, the problems encountered provide motivation and can inform the design of successful meeting capture systems.

2.1.1 Note-taking

The primary way people remember what occurs during meetings is by taking notes, writing down important items that will act as a memory aid. Notes tend to highlight important facts, actions, ideas, decisions and summaries. Note-taking has been studied mostly in academic settings, however Khan (Khan, 1993) studied note-taking in meeting settings in order to provide guidelines for building pen computers. Several meetings were observed, where he found that, on the whole, the frequency and length of the notes was very short. Actions were more likely to be referred back to than any other category of note. Notes were mostly organized in chronological order, and people found information by just flicking back through their pages of notes. The usefulness of notes appears to be very short-term, as many people said they rarely referred back to their notes, and only 1/3 reviewed them regularly. People reported overall satisfaction with their note-taking and no desire to change their strategies, yet 70% reported that there had been occasions when they wished that they had written better notes.

Whitaker et al. (Whittaker, 1994) further interviewed these people about the problems encountered with note-taking during meetings. The major difficulties they found were failure to note facts that turned out to be crucial later, illegible names, not enough time to write everything, reduced ability to participate, and the inadequacy of notes for later detailed understanding. A meeting capture system has the potential to overcome some of these difficulties without requiring changes to individual's note-taking. However, this study also implies that people are relatively satisfied with their note-taking and cope with the problems encountered and may resist trying other methods without immediate benefit.

2.1.2 Design Rationale

From a knowledge management perspective, meetings can serve as a valuable source of tacit and informal information. They could be searched and mined to categorize and find relevant knowledge that currently goes undocumented. Conklin (Conklin, 1996) and Selvin et al. (Selvin, 2001) propose methods for capturing this informal knowledge in meetings. These methods are strongly based on design rationale techniques, and thus, suffer the same problems.

The design or decision rationale community has investigated methods for recording more specific information: the assumptions and rationales behind design decisions as they are discussed and debated. Research in design rationale has focused on languages for describing the reasons and assumptions behind the design and associated methods for recording rationale. Despite apparent acceptance of the importance of rationale, these languages and methods have not been widely used. First, even with tool support, rationale is too time-consuming to record and difficult to maintain. Second, existing design rationale techniques are unable to capture the full richness and complexity of rationales. As Gruber and Russel state (Gruber, 1996), users of rationale need to construct or infer rationale from a variety of information sources, not look up answers in pre-defined structures as these techniques dictate.

In automated meeting capture, we are concerned with a broader set of multimedia information. Yet we can still learn from the problems developers encountered in trying to record design rationale. First, meeting capture cannot add too many burdens to meeting participants. Indeed, ubiquitous capture is meant to be fully automated and nearly invisible, though it will undoubtedly change some practices and require effort for some practices. Second, captured information should not exist in isolation as a pre-defined structure. Instead, the information needs to be integrated within the context of the entire work process. We can do this by capturing specific work activities and artifacts in meetings.

2.1.3 Meeting Conferencing Systems

There are several commercially available meeting support or conferencing systems, available to help people conduct synchronous meetings. Some of these provide recording capabilities. One common conferencing system is Microsoft's NetMeeting (http://www.microsoft.com/windows/netmeeting/). Organizations maintain NetMeeting servers to which individuals can connect to find and join meetings. The clients provide chat, a shared whiteboard, application sharing, and even audio and video transmission. IBM's Sametime (http://www.lotus.com/products/product3.nsf/wdocs/homepage) and Webex (http://www.webex.com/) are similar, competing products. All are beginning to add the capability to record what occurs in a meeting. However, the playback of that recording is currently just envisioned as a video-tape like replay.

The now defunct Mpath system from Smart Technologies (http://www.smarttech.com/products/mpath/index.asp, 2002) was another conferencing system that added work-specific artifacts such as agendas and decisions that can be edited and saved during a meeting. Again, however, the record and replay capabilities were similar to a video tape.

2.2 Ubiquitous Capture and Access

Ubiquitous computing has as one theme the capture, integration, and access of everyday activities, creating a multimedia record for later perusal. Computers that we carry with us, and those that are in the environment, can be instrumented to do what they do best, record information. This leaves humans free to do what they do best, actively engage in and synthesize what is going on around them, without worrying about tediously preserving details for later memory. Indeed, many of the visions of the future role of computing include aspects of computers automatically recording the important details of our lives and making available those details when we need or want them (Abowd, 2000).

Truong et al. (Truong, 2001) define capture and access as the task of preserving some record of live experience for potential review some time in the future. Humans already attempt to capture events using many methods, including taking notes, drawing diagrams, creating memory aids, or snapping pictures. Automated capture is when a computer records a history of some activity. The recorded artifact is a stream of information that flows through time (Cruz, 1994; Brotherton, 1998). Some multimedia artifacts, such as video composed of 30 frames a second, are naturally such streams. Other activities can be recorded through a series of significant events. An access application provides views of the recorded streams of information related by time and playback of the recorded artifacts and significant events. For example, an application could record the annotation of a document, then replay that activity by redrawing the marks over time. Time does not have to be an explicit component of the stream. For example, a notebook can be used by a human to take and review notes. Time is implied by the order of the text. However, without an explicit notion of time, those notes could not be integrated with other streams or replayed.

Capture and access systems have been researched for the past decade. Early uses of ubiquitous capture investigated improving ways of indexing and replaying audio streams. More complicated systems have been built for recording note-taking, educational lectures, and meetings. We use the indexing categorization from Section 1.2 to discuss noteable capture and access prototypes. However, as capture and access technologies advance, systems are beginning to combine a number of indexing methods to provide a variety of indices. Thus, several systems are mentioned more than once in the following discussion.

2.2.1 Explicit indexing

As introduced in Section 1.2, *explicit online* indices are pointers into the record that are created on the fly during the activity while users interact with the system. In other words, software provides support and capture services. As users interact with these services, their interactions are time-stamped and stored as indices. For meetings in particular, these indices can be created based on *user intention* or *interaction with artifacts*. Explicit offline indices are created through user activities in reviewing the captured record. However, little work has been done in this area, so the following background is limited to explicit online indexing.

2.2.1.1 Intentional indexing prototypes

There are many actions users may do during an activity to influence its capture. However, certain capture prototypes provide actions that are strictly for users to mark interesting places they may want to visit again. These kinds of intentional indices can be compared to bookmarks in today's web browsers. Early meeting capture systems supported this notion of bookmarking, as other interactions were difficult to record. SoundBrowser (Degen, 1992) was a modified handheld tape recorder that provided users the ability to bookmark an audio recording through multiple buttons on their recorders. Users could then individually determine important points in their activity and mark them as they wished, the goal being to allow them to more easily find the places they wished to review later. Hindus and Schmandt (Hindus, 1992) present a capture system that allows people to mark interesting portions of an ongoing telephone conversation. Dynomite (Wilcox, 1997) allows users to attach keywords to their notes. Bookmarks can be pre-defined to belong to certain categories, such as "to-do item" or "important topic", or be defined by the user.

Besides just timepoints, systems are incorporating a "snap" or "screenshot" that creates an image of the computer screen that is linked to the rest of the captured record. For example, the MeetingCompanion product by Quindi (http://www.quindi.com/) runs on a PC and allows users to manually take a snapshot of the screen, or of the environment using a Webcam, at any time. Similarly, Workspace Navigator (Ju, 2004) records activity in a dedicated design space, and also allows users to take a snapshot of the computer screen in that space.

2.2.1.1.1 Video annotation systems

While capture systems have been built to record activities as they occur, the act of bookmarking, or categorizing, the recording is similar to video annotation systems built to aid qualitative researchers. Experimenters record an activity to study, then do detailed analysis of this video tape to pull out interesting phenomenon, transcripts, detailed behaviors and interactions. Video annotations systems have been proposed to aid this task by allowing the experimenters to create indices into this video to mark significant places to return. Such systems also focus on the ability to synchronize handwritten or typed notes with the video. For example, EVA (Mackay, 1989), VideoNoter (Roschelle, 1990), and U-Test (Kennedy, 1989) allow experimenters to customize keyword or event buttons that create indices when pressed, as well as type notes. Marquee (Weber, 1994) allows users to create time-zones, and attach notes and keywords to that zone. This allowed users to annotate sections of video, both in real time, and after the fact. Marquee designers found that users were able to take notes using these time-zones. These systems could be considered examples of explicit, offline indexing of recorded material. However, little work has been done in examining users' behavior in interacting with these systems. Thus, they can provide possible suggestions for capture prototypes but do not help understand the use or impact of such systems.

2.2.1.2 Interaction with Artifacts

The most common way to explicitly gather indices is through recording user interaction with artifacts. Artifacts generally are *persistent* objects that relate to the work people are doing. Most existing meeting capture systems have focused on a small set of general artifacts to be recorded, such as slides, notes or whiteboard activity.

2.2.1.2.1 Note-taking

Several systems have explored augmenting personal note-taking with audio recordings. In these cases, hand-written notes are automatically time-stamped, which then later serve as indices into the audio recording. The AudioNotebook (Stifelman, 1992; Stifelman, 2001) is an example of augmenting a physical notebook with a microphone for recording, and speakers and an LED display for playback. The advantage of using physical paper is that the medium for taking notes does not change, but is instead truly augmented. The disadvantage is that the playback may be harder to visualize and control with only an LED and simple buttons.

Dynomite (Wilcox, 1997) is a digital notebook for taking notes and linking them to audio. Dynomite also examines how to structure, index, and search those notes for more flexible access and review. Dynomite users take notes on a tablet computer. As mentioned earlier, they can classify notes with properties, such as “To Do,” or “Name” before or after writing the note, or add keywords to any page of notes. Finally, audio is constantly captured, and snippets are stored and indexed permanently when indicated by the user.

Several systems have focused on note-taking during meetings. Filochat (Whittaker, 1994) was targeted towards meetings, but closely resembles a digital form of the Audio Notebook, integrating hand-written notes with audio. A new page of notes contains a header for the topic of the meeting and attendees, but could be used to record any type of discussion. Users control the playback of the audio with standard play, forward, and back buttons as well as gestures at specific notes to position the playback.

LiteMinutes (Chiu, 2001) focuses on a minute-taker typewriting notes. Each line of notes is then synchronized to the audio and can be conveniently emailed or posted to the participants of the meeting.

Notelook (Chiu, 1999) is a descendant of Dynomite that explores augmenting personal notes with information from meeting rooms, specifically video. Notelook allows users to incorporate snapshots of the meeting room or of the presentation into their notes and further annotate on top of them. These notes are also synchronized to the video of the meeting.

The commercial product MeetingCompanion by Quindi (http://www.quindi.com/) allows users to create personal meeting records. MeetingCompanion runs on an individual PC, and can record audio or video, images of the user's desktop or slides, and general notes. Lines or chunks of text create an index into the record. Additionally, the interface supports the creation of keywords, such as agenda items or questions that can then be easily inserted into the notes as they occur. While the features of MeetingCompanion sound like a combination of Notelook and LiteMinutes, this system is also the most similar in concept to our TeamSpace system, focusing on natural explicit indices and audio.

2.2.1.3 Slides and electronic whiteboards

The note-taking systems mostly focused on capturing personal information. However, other systems capture the public artifacts of a meeting or lecture, mostly focusing on the slides and whiteboard.

DUMMBO (Brotherton, 1998) supports informal, opportunistic meetings that would take place around a public whiteboard. The system monitors an electronic whiteboard for activity, and begins recording the whiteboard activity and audio when someone starts to write. Additionally, users could use electronic badges or other sensing technology to notify DUMMBO who was present near the whiteboard. Users could browse the activity through a Web interface, filtering for who, when, and where a meeting might have taken place. The whiteboard activity could then be played back synchronized with the audio.

A more structured environment, the We-Met (Wolf, 1992) system tried to support effective group communication in meetings, as well as capture and review of meeting information for small, informal work groups. We-Met consisted of a pen-based drawing environment, where users could create and manipulate shared "scenes" for drawing or taking notes. All system events were recorded and time-stamped, and could be replayed during or after a meeting.

In the education domain, annotated slides are also important. For example, eClass runs in an augmented classroom where video and audio are automatically recorded. Additionally, the lecturer uses eClass software on an electronic whiteboard to show and mark slides. Both the slide changes and annotations serve as indices into the audio or video.

Authoring On the Fly (AoF) (Muller, 1998) also aims to support the creation, distribution, and recording of distributed presentations. With many drawing and editing features, AoF supports a wider range of presentation activities, such as annotating the slides, viewing and annotating additional materials such as images, and drawing on a whiteboard. The interactions with the AoF whiteboard are all played back synchronized with the audio and video of the presenter. However, AoF is focused more on creating distributed content rather than on merely capturing current meeting activities. Thus, there is more effort involved in learning how to use the system and in putting together the presentations.

2.2.1.4 Domain specific activities

Other capture systems have focused on more domain-specific artifacts to capture. While the Tivoli system (Moran, 1997) did focus on electronic whiteboard activity, they more broadly addressed the creation and manipulation of content on large whiteboards. Tivoli enables people to use gestures to create, edit, manipulate, and relate objects on an electronic whiteboard. Domain objects, such as cost items or budget center objects, could be created and specialized behavior associated with specific gestures. These objects are then stored for later retrieval, with the gestures synchronized with the audio recording for playback. However, these objects represent fairly small chunks of content.

2.2.2 Derived indices

Creating derived indices has been another popular research activity, addressing needs beyond just meeting capture and access. We defined d*erived* indices as implicit, sensed, or inferred through other means and not being related to direct user actions. Much of the work has focused on audio, video, and slides, but also supports handwriting and text. Analysis of media streams or other unstructured information can be done either online or offline depending on the processing power available, the algorithms used, and the intended use of indices (some of the indices might be required while the meeting is still going on).

There are many different kinds of analysis that can be done on audio streams, such as pause detection, speaker identification, speech recognition and keyword spotting. For example, Hindus and Schmandt (Hindus, 1992) use pause detection and other analysis to detect turn-taking in a phone conversation, and thus allow indexing based on the speaker. Speech recognition is a major field of research, and further text analysis can be done following speech recognition. For example, the Jabber system (Kazman, 1996) uses WordNet and lexical analysis to create indices based on the discussion concepts.

Video analysis techniques can be used to determine and create a number of indices. One common use is to determine slide and scene detection. The LectureBrowser (Mukhopadhyay, 1999) application had a similar intent to eClass, however it accomplished classroom capture using video cameras. Video techniques were then used to deduce slides and slide changes to provide indices into the recording. The Portable Meeting Recorder (Lee, 2002) uses a variety of video analysis to determine room location by matching the background image against room templates, perform face detection to gather good shots of meeting participants, and to determine significant visual events. The Interactive Systems Lab at Carnegie Mellon are investigating a variety of multimodal techniques to perform speech recognition and participant identification, dialogue analysis, meeting summarization, recognize action items, and detect focus of attention and various speaker properties (Bett, 20000; Schultz, 2001).

Finally, ink can be interpreted as text using handwriting recognition or as diagrams using sketch-based recognition. Hammond et al. (Hammond, 2002) capture design meetings where freehand sketches of UML diagrams are recognized and the interactions in the creation and manipulation of those sketches are time-stamped and serve as indices into the recording. This is a similar idea to our SAAMPad system, but relies on deriving the diagram indices as opposed to our explicit method.

Deriving indices from media streams involves less intrusion into the activity being recorded. Capture is then more invisible to the user, requiring no change in their activity and potentially easing adoption of the capture services. Researchers have made tremendous advances in recognition technologies in the past decade. However, determining fine-grained and semantically meaningful activities is extremely challenging and will require significant research.

2.3 Visualization & Review of Information

Thus far, we have focused mainly on the capture aspects of existing capture and access applications. Many prototypes provide fairly simplistic playback of the recordings showing the video, slides and a simple timeline, or simply initiate playback in the same interface used for capture. A few, however, have created or suggested more complicated interfaces and visualizations for the playback of recorded information. For example, Manga (Uchihashi, 1999) summarizes meeting video with a comic book style image summary based on detected keyframes that can be annotated with captions of meeting minutes. Users can then navigate to the full video using those images.

Eclass focuses its review around the slides, with an integrated timeline showing both in-class slide and web page visits. The system interfaces are very robust and refined as they have supported thousands of lectures and hundreds of students. Web pages for each course are automatically generated with listings of each recorded lecture, and pages can be personalized to show or hide different portions of the interface and information. A set of search pages allow students to search for keywords in slide text and web pages. Additionally, a timeline-based interface for browsing larger sets of classes was experimented with. The extensive use of eClass implies that successful access applications will not only involve simple playback of the recordings, but also other features for customization, searching, and using media.

In Tivoli, researchers created a "salvage station" to assist a user in writing reports from recorded meetings (Moran, 1997). The user had the pages of the whiteboard activity along with typed notes from the meeting, a timeline and playback buttons, and a word processor for creating the reports. The user could also use Tivoli to create additional annotations and indices during playback. Thus, the Tivoli tool was not only used to capture meeting activity, but was used during playback to further mark the recording. Meeting access interfaces need to enable users to further organize and structure the meeting information, and create additional interpretations and summarizations.

Ginsberg and Ahuja (Ginsberg, 1995) discuss an interface for reviewing histories of virtual meetings. They represent the histories as a timeline with particular events represented as icons. In their visualization, the activities of each participant are shown in a row, with icons representing audio recordings, mail messages, or joining and leaving the environment, and colors showing event duration. The visualization can scale to longer time frames or more participants by compressing the timelines into fewer pixels. This is a similar idea to the visualization in our own TeamSpace system. However, this interface would require a large amount of screen space, thus limiting its usefulness for navigation while reviewing other information. Additionally, Ginsberg and Ahuja are unclear what tasks they are attempting to support with this visualization. We feel that one timeline for the overall history of the meeting would be more appropriate, not the history for every individual participant. Ginsberg and Ahuja do not seem to have implemented this visualization as part of any real access interface.

Outside of capture and access, visualizations have been done for other problems involving time-based material that could be useful for visualizing meetings and their indices. Lifelines (Plaisant, 1996) is a visualization for displaying a large amount of time-based personal information. The visualization has been used for personal medical histories among others. Lifelines consists of a timeline, with time-based information such as lab tests or doctor visits shown in different rows along the timeline, and represented by different colors. Each row can be represented in just a few pixels, or can be expanded to add labeling and more information. Each "event" can be opened to view the details, such as viewing the doctor's comments or an x-ray. The visualization for the review of meetings was based on the lifelines idea. However, in a meeting, each event does not necessarily correspond to a detailed piece of information. Thus, the amount of pixels for each event, and the labeling involved has been greatly compressed. Additionally, lifelines are not meant to be "replayed." However, this visualization may be an excellent way to represent the history of the entire project, involving many different kinds of documents, schedules, meetings, and other events. Examining this is beyond the scope of this thesis, however.

2.4 Use of Capture and Access

Despite over a decade of research in capture and access, we have seen relatively little use of capture and access prototypes. The exception to this is in education, particularly educational seminars in organizations. AutoAuditorium (Bianchi, 1998) is a commercially available product for producing higher quality recordings of formal presentations. He et al. (He, 1999; He 2000a; He, 2000b) have studied review of video summaries of informational presentations within Microsoft. One study focused on comparing three techniques for auto-summarization of presentations, such as generating summaries based on slide-changes or speaker pauses. They found that users feel that computer-generated summaries are less coherent than summaries produced by authors. However, they do grow accustomed to these summaries and do not seem to prefer any particular computer-based method. Another study compared reviewing a presentation with slides only, a text transcript, the transcript with highlighted points, and a video summary created by the author. Users had a marginal preference for the video summaries. Styles of slide-authoring had a large impact on the effectiveness of the different kinds of summaries. Finally, they evaluated the extensive usage data from Microsoft's recorded presentations to examine real playback usage patterns. They found that in general, the number of viewers of any individual presentation segment decreases as time progresses. In other words, users tend to start watching a presentation at the beginning and stop sometime before the end. Additionally, the number of viewers spikes around slide changes, implying that users are navigating based on slides.

The evaluation of eClass is the most extensive evaluation of a capture and access system to date (Brotherton, 2001; Brotherton, 2004). The system was used on a daily basis, in multiple classrooms, to record over 2000 lectures. Logs of over 50 thousand accesses to the notes have been analyzed, along with hundreds of student and teacher questionnaires. The result is a deep understanding of how eClass is really used when it becomes part of the normal lecture activities. The amount of logs shows the overwhelming use of eClass. Students reported that it was a useful study tool, and their usage indeed shows that much of the access occurred prior to tests. Yet the system was used beyond that, with many accesses occurring after the semester's end. The system had no negative impact on note taking, seeming to encourage more summary style notes. Augmenting the notes with audio and video was useful, as nearly 20% of study sessions accessed this media. Accessing the media by slide indices was the most popular, even though accessing by ink and by a generic timeline was possible. Students reported that they preferred classes with eClass. However, they desired the ability to take their own notes. While the evaluation did not show any positive impact on test scores, it did show no negative impact and students reported that eClass helps them learn, succeed in class, and be more efficient in studying.

The challenge in meeting capture and access is that there is a less obvious need for review. In the classroom, students have a real need to review and study for exams. In an organization, many people will not attend interesting presentations, and thus the recordings serve to inform people when needed. However, meetings vary greatly in content, importance, and need for review and individuals' motivations and willingness to invest time in reviewing differ as well.

Most of the meeting capture and access systems built have been evaluated on a small scale, showing their usability and unobtrusiveness during the meeting. A few looked specifically at the use of recordings. Wolf (Wolf, 1992) studied the retrieval of meeting information through videotaping 10 meetings. Afterwards, participants were asked to recall meeting details, as well as find specific information in the video-tape. People searched the video using a number of characteristics, such as participant behavior, the communication medium, time, and the relation to other events. Participants used a "sampling" technique, viewing small segments, fast forwarding, then viewing another segment. This study suggests that a number of semantic cues can aid people in finding information in meeting recordings. Making some of these cues explicit may help people find information faster, which we aim to do in examining indices based on meeting activities. Similarly, Kazman (Kazman, 1996) examined the effectiveness of having video retrieval in Jabber over memory alone. They asked subjects questions from a previous set of meetings, and found that using the video substantially improved their responses, especially for design decision and design rationale information.

Wittaker (Whittaker, 1994) observed users of Filochat in 9 meetings. The main stated benefit was that it allowed the generation of higher quality minutes. They also conducted a field study using pen/paper, a Dictaphone, and Filochat to compare the accuracy, efficiency and confidence of different note-taking mediums. Filochat gave participants the most confidence and the most accuracy in answering questions, but at the expense of added time. Thus, there seems to be a trade-off between access time and accuracy of recall.

Stifelman (Stifelman, 2001) performed a longitudinal field study with the Audio Notebook, observing four students and two reporters performing real tasks over a five-month period. The study showed that users had different ways of using the notebook. For example, one student did not change her note-taking style, but instead complemented her review by being able to listen and skim the audio. Another student began taking fewer detailed notes, and relied more heavily on the audio for review. This implies that capture and access services need to support multiple strategies of use for different people and tasks.

Ju et al. (Ju, 2004) also deployed Workspace Navigator to sets of student design teams. They found that the teams did make use of the system to incorporate work into their final semester reports. They also noted the benefit of implicit capture as one of the groups did not attempt to use the captured material until the end of the semester when they realized that they had not properly documented their design process themselves and that the system provided them with that needed information.

The most extensive study of the use of meeting capture and access is from Moran et al. (Moran, 1997) who observed one person using Tivoli to aid in writing reports of intellectual property meetings. The participant used a “salvage station” at his desk to review the meetings, extracting and organizing information in order to write the reports. Also, the user could create further annotations and indices during review. Moran found that the user employed several salvaging strategies to prepare the reports, which changed over time. Initially, the user mostly wrote the report while listening to the captured audio and reading the notes. He listened in a mostly linear fashion to the recording. Over time, however, this strategy changed. The user was more focused, listening to only small portions of the audio, which he often transcribed and put directly into the report. He started using special marks during the meeting to help him find these relevant sections. He also made further annotations and indices while reviewing. However, the user seemed to use this focused salvaging strategy more often for difficult and unfamiliar materials, still using the more direct playback for topics he was familiar with. Moran found that besides just reviewing the recording, the user needed a workspace to collect and organize relevant information from multiple sources. He did this by adding notes to the recording, but an explicit workspace may have been beneficial. The implication is that tools need to provide an explicit workspace for collecting and integrating information, both from the recording and other sources. Another conclusion from this study is that users need ways to manage, structure, and control the salvaging process itself, such as with better tools for navigating and controlling the recording. This study also demonstrates the evolution of behavior that occurs as a user experiences and adapts to capture and access services. This only underscores the need to encourage repeated usage of capture services to enable many more long-term evaluations.

2.5 Summary

Much research has been done in capture and access, and even in meeting capture and access. Research in note-taking and meeting memory show the potential benefits of capturing meetings in more accurately recording information that may otherwise be lost. Many prototype systems have been built. Yet none have been put to real use at even a fraction of the scale under which eClass was used and evaluated. Lessons learned from design rationale may provide one answer: captured material needs to be related to the rest of the context of work. Additionally, not enough research has been done on the actual replay and review of meetings to accomplish real work tasks. Meeting capture capabilities are even being added to commercial conferencing software, yet without understanding how these recordings can be utilized, they will be no more useful than traditional videotapes of meetings. Because eClass helped students fill a real need, reviewing lectures and studying for exams, the system was able to be adopted into real, everyday practices of students. The report writer using Tivoli also had real needs to review meeting information. We need to also find groups of users with real needs for recorded material to encourage the usage of our meeting capture prototypes and perform similarly successful evaluations.

CHAPTER III

CAPTURING ARCHITECTURAL RATIONALE

We begin our capture and access explorations with a discussion containing important structured content that needs to be supported and indexed. In software development, numerous artifacts are created, discussed, and reviewed in many different types of structured meetings. As an initial exploration, we focused on a type of design evaluation meeting, namely the Software Architectural Analysis Method (SAAM). SAAM (Kazman, 1996) is a structured method for understanding the high-level organization of a software system and determining the impact of requirements changes on that structure. The method revolves around group discussions by the various stakeholders in the system. With two prototypes, we explored anchoring the capture around the specific artifact in the meeting – namely, a software architecture. We performed an early evaluation of the access interface in finding information. Our results show that structuring capture and access around such a specific artifact is indeed possible and understandable, and that recording such meetings may be valuable and useful.

3.1 SAAM

The Software Architectural Analysis Method, or SAAM, was developed at the Software Engineering Institute (SEI) in the mid-90s to enable software developers to compare different proposed architectures based on how they would be impacted by current and future requirements of the system. However, the same technique may be used to determine how an existing system may be affected by evolution. The SAAM method revolves around group discussions by the various stakeholders in the system, including designers, customers, and users.

Software architecture in SAAM refers to the components into which a system is divided at a gross level of system organization, and the ways in which those components behave, communicate, interact, and coordinate with each other. Components may be processes, tasks, or classes. These components may communicate via system calls, message passing, or event broadcasting. Communications are sometimes referred to as connectors. SAAM attempts to determine the quality of the architecture, or in the case of evolution, the impact of the evolution on the design.

Quality of a system can only be measured with respect to some attribute. SAAM uses scenarios to express particular quality attributes or new system behaviors. The analysis team then discusses how well or how easily the architectural design satisfies the demands placed on it by each scenario. SAAM's scenarios are brief descriptions of some anticipated or desired use of a system. One example might be "users can change the color of the window borders." Scenarios can differ widely in breadth and scope. Scenarios should also include all the different roles involved in a system, such as the user, the operator, and parts of the software.

The meeting is structured into the following steps:

1. Describe the existing architecture. The architectural descriptions need to be understandable by all parties involved in the analysis. They need to include the system's computation and data components, as well as all the connectors.
2. Develop scenarios. Develop task scenarios that illustrate the kinds of activities the evolved system must support and the kinds of anticipated change to the system
3. Perform scenario evaluations. For each scenario, determine whether the

architecture can execute it directly, or whether a change would be required to execute it (which we refer to as an indirect scenario.) For each indirect scenario, list the changes to the architecture that are necessary for it to support the scenario and estimate the cost of performing the change. A modification to the architecture means that either a new component or connection is introduced or an existing component or connection requires a change in its specification.

4. Summarize the information. Summaries could include tables of components and scenarios and the set of changes required for each, or diagrams highlighting changed components.

3.2 SAAMPad

During a SAAM session a great deal of architectural rationale can be discussed. Constraints and assumptions may be raised while understanding the original architecture. Scenario evaluation may involve considering a number of solutions and making tradeoffs. Yet users trying to take detailed notes of all of this information are not likely to fully participate in the discussions. Additionally, it is hard to know exactly how important certain items are during discussion so that they may be documented more fully. Instead, they only become important at the time they are actually needed. As such, we proposed a new way of automatically capturing the entire experience of a SAAM session and allowing easy access to the information later.

The approach we used to achieve these goals was to start with a general capture tool and extend it to support SAAM specific activities. As part of his Master's thesis, student Pascaal Schuchhard implemented the capture system SAAMPad using Zenpad,

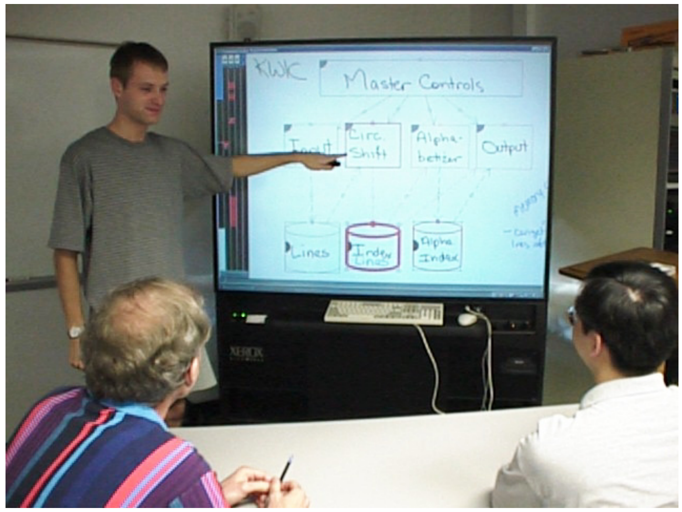

Figure 3.1. The SAAMPad system.

the central tool in the eClass system, as a basis (Richter, 1999). Zenpad (Abowd, 1999) supports the capture of audio and video streams as well as annotations made on an electronic whiteboard. The SAAMPad extensions involve providing specific behavior to support SAAM activities, such as drawing an architecture and marking changes, and supporting different behaviors for the different phases of the SAAM process, see Figure 3.1. Additionally, we created visualizations of SAAM sessions to facilitate the review and retrieval of the recorded information around the interactions with the software architecture. Despite having a working capture tool, we did not have many real situations where we could use the tool, nor did we understand how the captured information would be used by actual software engineers. The access needs had to drive the capture.

3.3 SAAMPlayer

Next, we concentrated on the review of real SAAM sessions. As a preliminary case study, we facilitated and video recorded a SAAM analysis with a group of designers at a telecommunications company. Their legacy system provides real time monitoring and historical reporting for an automated call distribution center. The system was originally developed by another company and had undergone numerous changes and improvements over many years. Thus, an architectural understanding of the system was distributed over a group of designers. They wished to use SAAM to come to a common understanding of the overall architecture that they could then use to discuss requirements changes. The analysis took place over three separate meetings. We facilitated and participated in the meetings, while another graduate student ran the video camera. The six company participants were responsible for designing and implementing the changes we discussed. Each meeting focused around a whiteboard for drawing architectural diagrams, and two flip charts for brainstorming. Following the meetings, the Georgia Tech participants prepared a detailed written summary of the results of the analysis for the company.

Next, we created a prototype interface for browsing and viewing the digitized video of the meetings. This interface, called SAAMPlayer, is shown in Figure 3.2. SAAMPlayer consists of a video window, simple playback controls, and a timeline for video browsing. As part of this effort, we created a general annotated timeline tool, the Multi-Scale Timeline Slider (Richter, 1999), which has been reused in other access prototypes (TeamSpace in particular). The Multiscale Timeline Slider provides a focus region that can be created on the timeline and cascaded into focused timelines. We decided that an electronic version of the summary document could serve as a natural index into the video, and could realistically be produced automatically with speech recognition technology and something resembling SAAMPad. We manually created

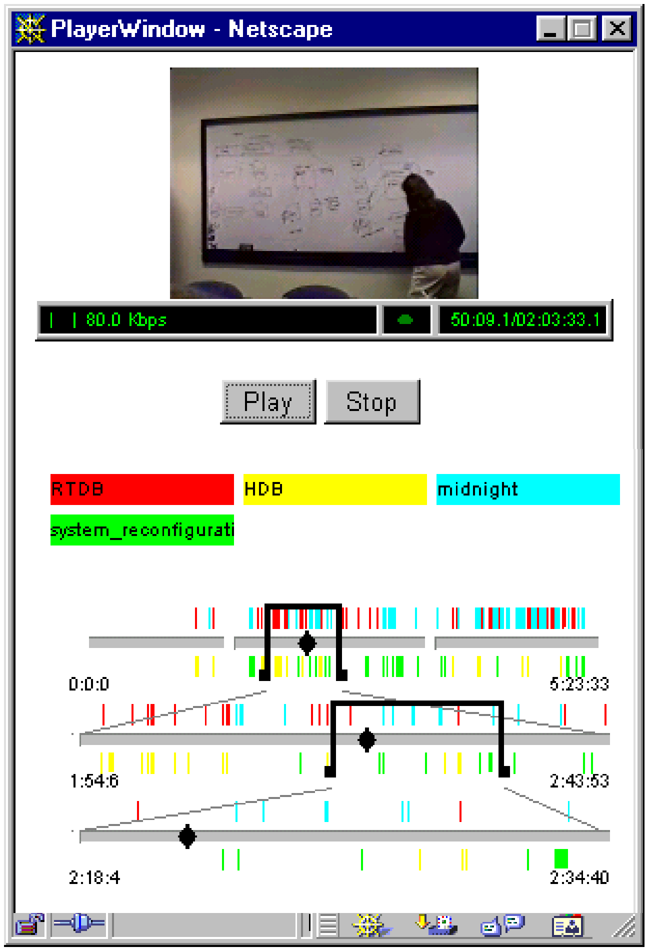

Figure 3.2. The SAAMPlayer interface. A multi-scale timeline is annotated with keyword locations.

timestamps for a set of keywords, namely the architectural elements of the system and several general issues that were discussed. When the user clicks on one of those keywords in the document, the timeline of the video is annotated with the points where that keyword was discussed. Browsing the video is then a process of using keywords to find relevant sections of the video to replay.

3.4 Evaluation: Access Behavior and Use

The affordances for navigating the video in SAAMPlayer are relatively few, namely time and keyword locations. As seen in Figure 3.2, the timeline is rather dense, with each focused timeline being less dense. Thus, while the timeline does provide some cues, these cues may not be specific enough to be useful.

We ran a small experiment to examine the use of SAAMPlayer and determine its effectiveness for accessing design meetings (Richter, 1999). We wanted to observe the facility with which users could find information in the video. There were four subjects for this trial: two subjects who were in the original SAAM meetings, and two subjects who had no prior knowledge of the architectural analysis. The subjects were given the SAAM summary document to review prior to the experiment. They were also given a brief tutorial on using SAAMPlayer. The experimental task was to answer a set of questions about the system architecture and the issues that were discussed in technical detail. They were instructed to use the document and the video as much or as little as they wished. However, the questions were designed so that the document obviously did not contain all of the details. For the subjects who were not in the meetings, the questions asked for details of certain system components, as well as explanations of key issues. The other subjects were asked more complex questions about a key issue that was not resolved in the meeting. We did not measure how well the subjects answered the questions. Instead, we used the questions to motivate the subjects to look at the video so we could concentrate on how the subjects browsed and found information in SAAMPlayer. We asked the subjects to think aloud when they were browsing the video. Following the task, users were asked for feedback on using SAAMPlayer.

3.4.1 Browsing Behavior

The browsing provided by SAAMPlayer is very simple. Still, successful strategies were employed that enabled subjects to find information. The first strategy was highlighting multiple keywords to look for areas where multiple components or issues were discussed. However, each user did this slightly differently. Figure 3.3 summarizes two aspects of this browsing. The x-axis represents the time elapsed during the experiment, while the y-axis is the number of keywords that were highlighted on the timeline. The lines for each subject are wider where they were playing the video. Subject 1 highlighted many keywords and looked for intersections between them. On the other hand, subject 2 only used one keyword. Subjects 1 and 4 stated that "being able to look at keywords that occur at the same time was valuable". The playback of the video also differed with each user. Subject 1 played relatively few segments. However, subjects 2 and 3 tended to skim the video. They would listen to one section for awhile, then skip ahead to a later section that appeared relevant. Despite his lack of skimming during the experiment, subject 1 suggested more automated skimming as an improvement to SAAMPlayer. He wanted to tell the player to automatically skip over sections that didn't contain any keywords he cared about.

The majority of the video segments that the subjects replayed were related to their tasks. Subjects 1, 2, and 3 found multiple passages they considered particularly interesting or critical. While other passages did not have as much impact, they still related to their search. However, there were cases where the segments were clearly not providing the subjects with anything. Subject 4 had difficulty starting playback at segments that contained the keywords she wanted. She spent several minutes making very slight adjustments to the scrub before finding the segments. On two occasions subject 3 became lost and accidentally replayed segments of the video he had just listened to. Not using a more focused timeline contributed to both of these problems.

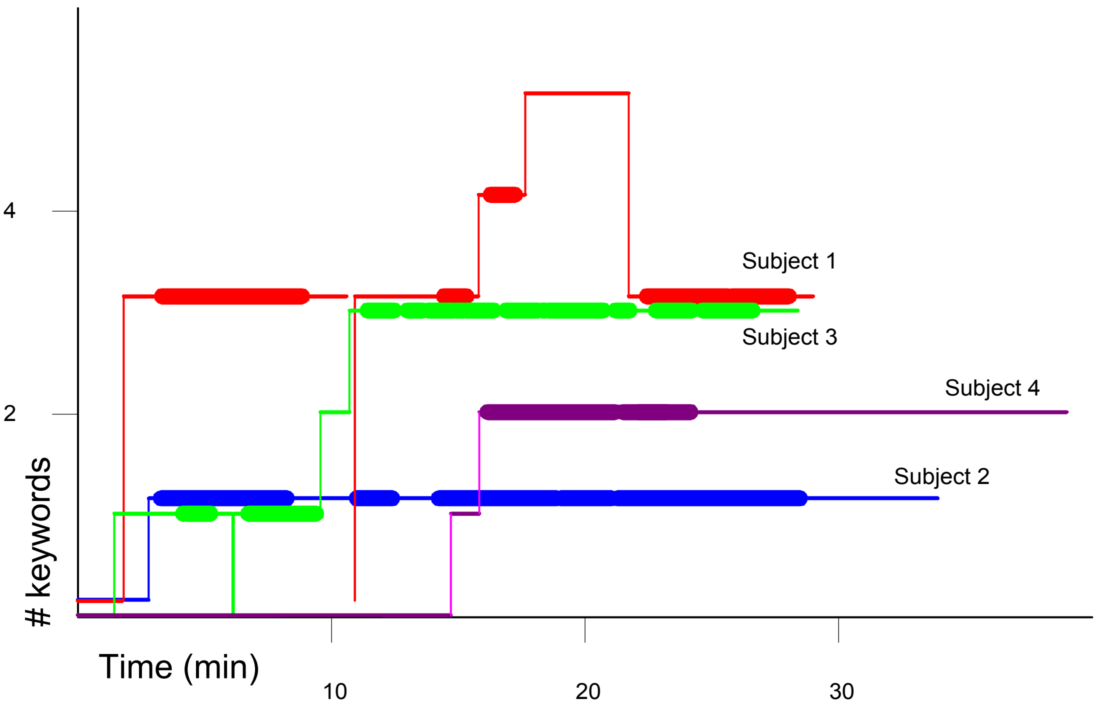

Figure 3.3. Summary of user browsing and playback.

We also need to look at whether the interaction with the tool was sufficient for subjects to complete their tasks. At first, several subjects had difficulty remembering how to create a focus region, which required a right-mouse-click. Subjects 1, 2, and 3 did create focused timelines and moved the focus regions as they browsed. They all controlled the video playback on the more focused region. However, subject 4 did not create a focused timeline and used only the top-level timeline for browsing. She later stated that it had not occurred to her to zoom, even when she was having difficulties playing back the right segments.

While the subjects liked using the keywords to browse, they all felt that more information would have helped them find important points more quickly. They wanted to know more about the activities of the people in the meeting, such as which stage of the analysis method the participants at. One subject wanted to know where keywords were defined. Another wanted to see where an issue was resolved. Other indices could be user-generated. One subject wanted to mark points he found particularly interesting.

3.4.2 Motivations of Use

In the above trial, the video was useful above and beyond the document in several ways. First, subjects were able to find new information that was not contained in the document. Subjects 1, 2, and 3 felt that they learned important details this way and emphasized this new information in their answers. Second, the video grounded subjects 1 and 2 in the meeting, sparking their memories about what occurred. Subject 1 spent several minutes listening to early portions of the video to remind himself of the system and the general issues. This, in turn, led him to choose different keywords to further browse the video. Finally, the video provided a different way of stating information that already was contained in the documentation. Subject 3 stated that he often finds documentation inaccurate. Thus, the video provided him more authority and confidence in the information he was reading. Additionally, the video often covered information at different levels of detail. Subject 2 stated that the document was sometimes too abstract or too detailed. He used to the video to find a different perspective of that information at a different level of detail. Thus, the video can be valuable for both providing new information and as an additional perspective or authority on already known information.

3.5 Discussion of Contributions

The prototypes built in this exploration were the first meeting capture systems to be structured around this type of specific, content-filled artifact. SAAMPad and SAAMPlayer demonstrate that these systems are feasible to build and use for capturing and accessing information. We believe our initial case study, while small and simple, illustrates the potential of such systems. Users were able to employ various browsing strategies to find valuable information, even with the limited navigation provided. The timeline, although compact and dense, provided enough indices to aid users in choosing and scanning areas of interest. The studies suggest possible improvements in the indexing of these meetings, involving more content-specific and user-initiated indexing which will be discussed in more detail in Chapter 6. The evaluation shows how recordings might be useful; subjects found the video useful above and beyond the standard document summarizing the SAAM session. What was surprising was that not only did the video provide additional information as expected, but it also provided an additional perspective or authority on already known information. Thus, the captured sessions can be useful even with complete written documentation available.

The challenge in more fully evaluating these prototypes is that SAAM sessions are relatively rare in a software development cycle, occurring perhaps once in a project's lifetime. Thus, gaining consistent use would be difficult, as each session would be the first, and likely last, time users would be exposed to such a capture and access system. Many of the interesting questions suggested by our first study would be difficult to uncover, such as: how much time would users be willing to invest in looking for answers, would people turn to the video as an additional authority, and what benefits would user-initiated context, such as bookmarks, add to the review? Additionally, the outcomes of the SAAM session are a shared understanding of the software architecture of a system. The benefits of using captured information to aid in recall of this understanding are difficult to measure and quantify. Thus, while these SAAM prototypes help illustrate our approach to capture and the unique kinds of indices we can leverage as well as the behaviors involved in reviewing information, we will not be able to evaluate the underlying motivation and benefits. In order to do this we turned to more capturing more generic meetings in order to examine a system in regular use.

CHAPTER IV

GENERAL MEETINGS

The workplace is filled with many types of everyday meetings. People share information, discuss ideas, report status, and review artifacts. While the content and structure of these meetings may vary greatly, there is still often commonality across many meetings. By taking advantage of this commonality, we can successfully capture a wide variety of discussions. The recordings of these meetings are likely to be less structured than more domain-specific discussions where we know more about the structure and expected content of the meeting. Yet by supporting a large amount of general meetings, we can aim to provide capture and access services that can be incorporated into regular, extended use. In focusing on these general meetings, we can explore many interface and use issues of building capture prototypes and utilizing meeting information. In this chapter we detail the design and implementation of a robust, deployed capture and access system. We discuss several evaluations of the system that focus on access behavior and motivations for use.

4.1 Artifacts and Activities

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 discussion types and meeting information, we interviewed three meeting facilitators of Boeing that work with multiple groups to improve or enable different types of meetings. Interviewee 1 was working on an initiative to set up an environment for cross-company collaboration. This involved a shared environment for communication and information storage. Interviewee 2 was creating a meeting architecture and electronic meeting rooms to support technical meetings. Interviewee 3 has been a team coach at Boeing, helping many different kinds of teams with their meetings. We asked them questions regarding the types and purpose of the meetings, information covered, and the technology used.

The meetings each discussed were very heterogeneous, both with respect to the people and topics involved. They involved teams called integrated product teams (IPTs) involving customers, suppliers, designers, numerous engineers, and manufacturers. At a lower lever were design-build teams (DBTs) involving different kinds of engineers and designers. Despite this heterogeneity, most meetings do have common and simple artifacts: agendas, action items, issues, presentations, and minutes. The most important aspect of these is the action items. In fact, they often serve as the most valuable minutes. Additionally, the only predominant technology for displaying all of this information is viewfoils (transparencies). They are prepared with whatever software is convenient, often Word® or Excel®, printed off, and brought to the meeting. Notes are sometimes taken on viewfoils as well, then copied and handed out as minutes afterward. Interviewee 2 was working on standard templates for some of these artifacts that could be filled in and used.

These interviews tell us that there do seem to be common meeting artifacts that are used and discussed, even across widely varying teams. These simple artifacts include agendas, action items, issues, presentations, and minutes — the most important of these would be action items for some groups. Second, there is no standard way to create and present most of these artifacts other than word processing software. Thus, we have a chance of getting users to use custom agenda and action item templates or tools if they add benefit. Finally, each of the interviewees thought capture had potential, but was unsure of how much.

As we discussed in Chapter 1.2, we are exploring capturing indices based on explicit interactions with artifacts within the meeting capture prototype. Based on our interviews, we decided to prototype our meeting environment with the following general meeting objects or artifacts:

- Meeting
- Agenda
- Action Item
- Presentation
- Document
- User

We then support common interactions with these artifacts, such as:

- creating and planning a meeting;
- editing and checking off agenda items;
- creating, editing, and assigning action items;
- changing slides and annotating presentations;
- changing pages and annotating documents;
- and inviting and checking attendance of users.

As users initiate these interactions during the meeting, time-stamped events are captured and later serve as indices into the meeting record.

4.2 Integration with a Group Workspace

We argue that for captured information to provide value, it needs to be related to the rest of users' work and everyday tasks. 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 implemented meeting capture as part of a larger team environment called TeamSpace, which was 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 (Geyer, 2001). The goals of TeamSpace are to support inter-company collaboration through awareness, information sharing, communication, and coordination. TeamSpace aims to support 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 envisioned are shown in Figure 4.1. 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 those activities, including those that are not captured and those that use captured information.

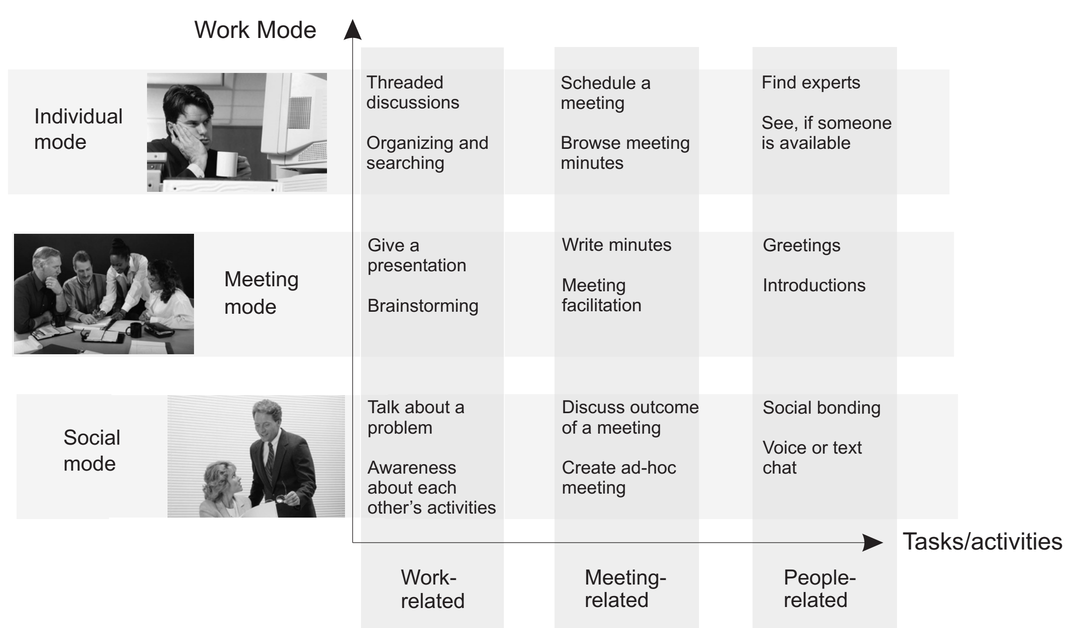

Figure 4.1. Work modes and activities.

In addition to supporting more meeting-related activities, integrating into an environment such as TeamSpace would 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 intended to provide mechanisms for relating meetings with other kinds of artifacts as they were incorporated into TeamSpace, such as more general documents or general project information. However, that extension of capture artifacts was not necessary for this thesis research.

4.3 The TeamSpace System

In the previous sections we highlighted how the desire for real-world usage led us to integrate our capture prototype within a larger work context. We now discuss the various aspects of the TeamSpace system and the features they provide (Richter, 2001). We further discuss implementation details in the next section. 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 support meeting capture, we created specific meeting-related objects based on the interviews highlighted in Section 4.1; namely, *Agenda, Action Item, Presentation, Meeting*, and *Person*. We designed, implemented, and deployed software to conduct and capture distributed meetings, and replay and review meetings. We attempted to design all of this software to provide general functionality, yet be flexible and evolvable so that more specific features could be added as we better understood its use by particular project teams. We also focused on reliability and consistency of the software and on intuitive interfaces as we wanted the system to be used and studied over a long period of time. Finally, we instrumented the software to facilitate understanding of users' interactions.

The need to gain as much information for as many users as possible led 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 by many people for multiple meetings. We also wanted to require as little instrumentation as possible to enable more locations for capture. The capture system was designed with an electronic whiteboard in mind, and desktop computers for distributed members, but can be run using just one laptop and an inexpensive microphone, if desired.

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.

4.3.1 Group Workspace Interface

Meeting preparation is accomplished using the web-based TeamSpace interface. From a series of tabs, users can create, edit, and view any of the objects that TeamSpace supports. 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. These other capabilities are less mature, and continued to be developed by Boeing after TeamSpace was initially deployed. We have not taken advantage of their full capabilities at Georgia Tech, or integrated those newer features with the capture and access capabilities.

Figure 4.2 shows the main interface, organized as a series of tabbed panes. The tabs are Home, Meetings, Documents, Action Items, and Project. The Home page summarizes the user's current objects, including upcoming meetings and unfinished action items. From the Meetings tab, users can see a calendar or list view of their meetings, search for meetings, and create and edit any meetings. Users can then join any meeting to launch the synchronous meeting tool, called MeetingClient, or review a completed meeting using MeetingViewer. The Documents section of the interface provides the shared workspace, where folders can be created and documents can be uploaded, tracked, and downloaded. From the Action Items tab, a user can see a list of his unfinished action items, and search for and track any other Action Items. To support multiple groups, all information is broken into Projects. Information can be viewed only by users in the same project; and users can be part of multiple projects. Additionally, throughout the interface, users can sign up for notification of various events such as when action items are assigned or due, meetings are available, or documents have been uploaded. Finally, a series of help pages give instructions for installing and using the various interfaces.

4.3.2 MeetingClient

The meeting capture phase is supported through the MeetingClient interface, shown in Figure 4.2. MeetingClient is launched automatically on a client's machine from the main TeamSpace web interface when joining a meeting. This client provides viewing,

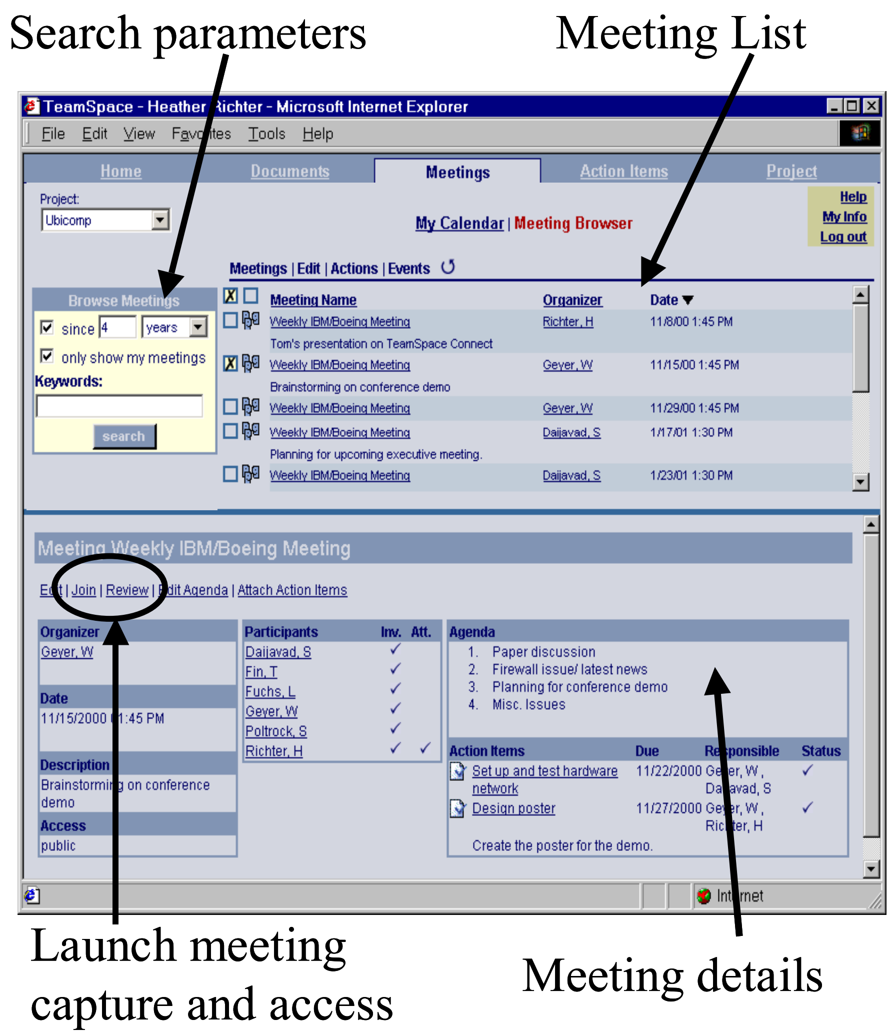

Figure 4.2. Screenshot of the TeamSpace group workspace interface.

editing, and annotation of agendas and action items, as well as viewing and annotation 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.

Additionally, users can intentionally create a "bookmark" event to create their own index into the recording. 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 4.3.3.

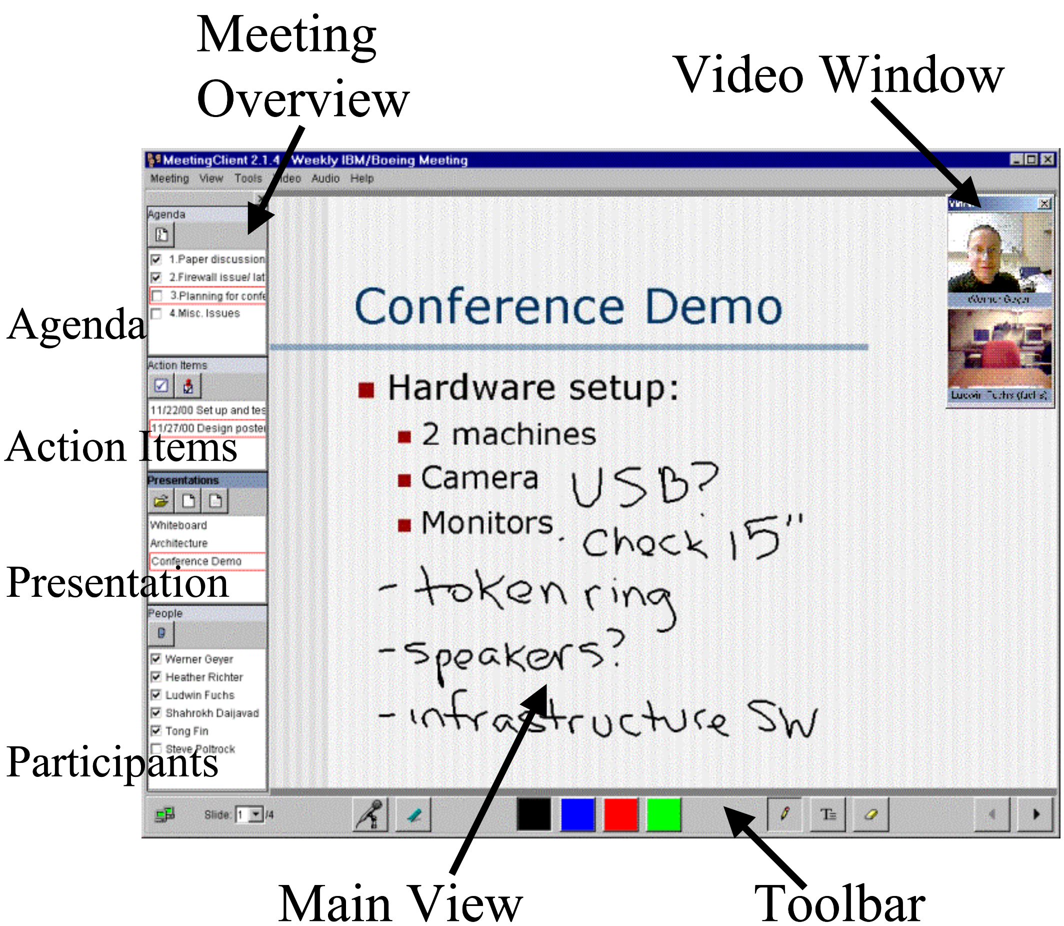

Figure 4.3. Screenshot of MeetingClient

The panel on the left of Figure 4.3 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 can be selected. Thus users can interact with the meeting objects, such as checking off agenda items and attendance, without changing the main view. 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, along with slide navigation and the bookmark and mute buttons. The interface was designed to work well on both pen interfaces, such as an electronic whiteboard in a meeting room, and on desktop machines. Thus, in the main view, agenda and action items can be typed or written in ink, and users can add both text and ink annotations to presentations. Blank presentations can be created to function as a whiteboard or notes page. All features available via buttons are duplicated in the various menus on the menu bar.

Audio recording is done using external software, RealProducer®, that is launched from MeetingClient on the client computer. Thus, only one client may be recording audio at any one time. If no audio is being recorded, the user is asked whether to start the recording when she joins the meeting. If recording is already occurring, the user is informed of this recording. Audio recording can also be started and stopped from menu options on MeetingClient. A mute button on the MeetingClient toolbar allows the audio to be muted, recording silence instead. Thus, private or sensitive conversations can still easily occur. Additionally, MeetingClient provides low-bandwidth video, which is viewed in a separate window, providing real-time awareness of other disbtributed team members. Video is captured using any basic webcam.

Any number of people can join in a meeting using MeetingClient. While this was designed to aid distributed meetings, multiple co-located users could also join to allow them all to interact with and annotate the meeting. 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, especially in the event of network delays. However, we wanted to keep the interface as simple as possible and investigate where synchronization through social protocols was not sufficient and what tool support could help manage the flow of the meeting. Most of the uses of MeetingClient thus far have been co-located. Thus the distributed meeting features, such as the video and shared data views, have not been extensively used or evaluated.

4.3.3 MeetingViewer

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 the MeetingViewer applet to view and playback these meetings.

The MeetingViewer, shown in Figure 4.5, integrates all of the meeting information based on time. The viewer uses a two-scale timeline (see Figure 4.4) for navigating a set of selected meetings, providing random access playback with finer-grained navigation on the lower timeline. The lower timeline shows a portion of the

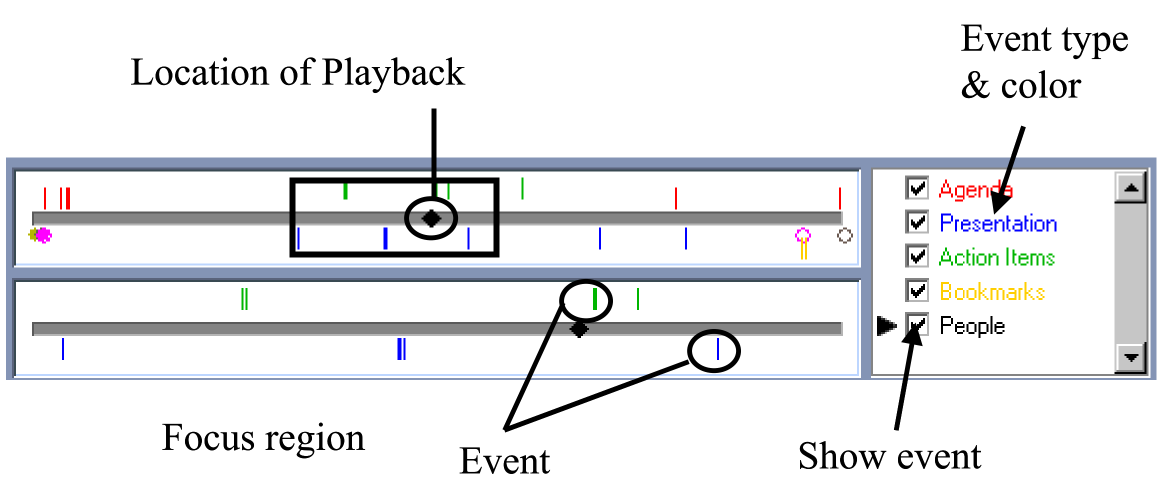

Figure 4.4. MeetingViewer timeline. The lines above and below the timeline represent interactions with the agenda, presentation, or action items. The circles are people join and leave events.

entire meeting, indicated in by the black box focus region in the upper timeline. The timeline is painted with color-coded events as both a visual summary of the meeting, and as an aid for navigation. Again, the events currently captured by MeetingClient are people joining and leaving, agenda items being discussed, action items visited or created, and slides visited, but the timeline could include any envisioned events such as people speaking and keyword locations should the capture capabilities evolve. Users can control which of these events they view and can use the events to find relevant portions within a meeting to playback. Additionally, users can click on the timeline or an event line, in addition to dragging the timeline scrub, to navigate the meeting. 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 TeamSpace evolves, 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.

The agenda item and action item views function similarly. Items are highlighted as they were highlighted during the meeting, to show that they were the topic of discussion. Small target-shaped icons can be clicked to jump to the point in the meeting

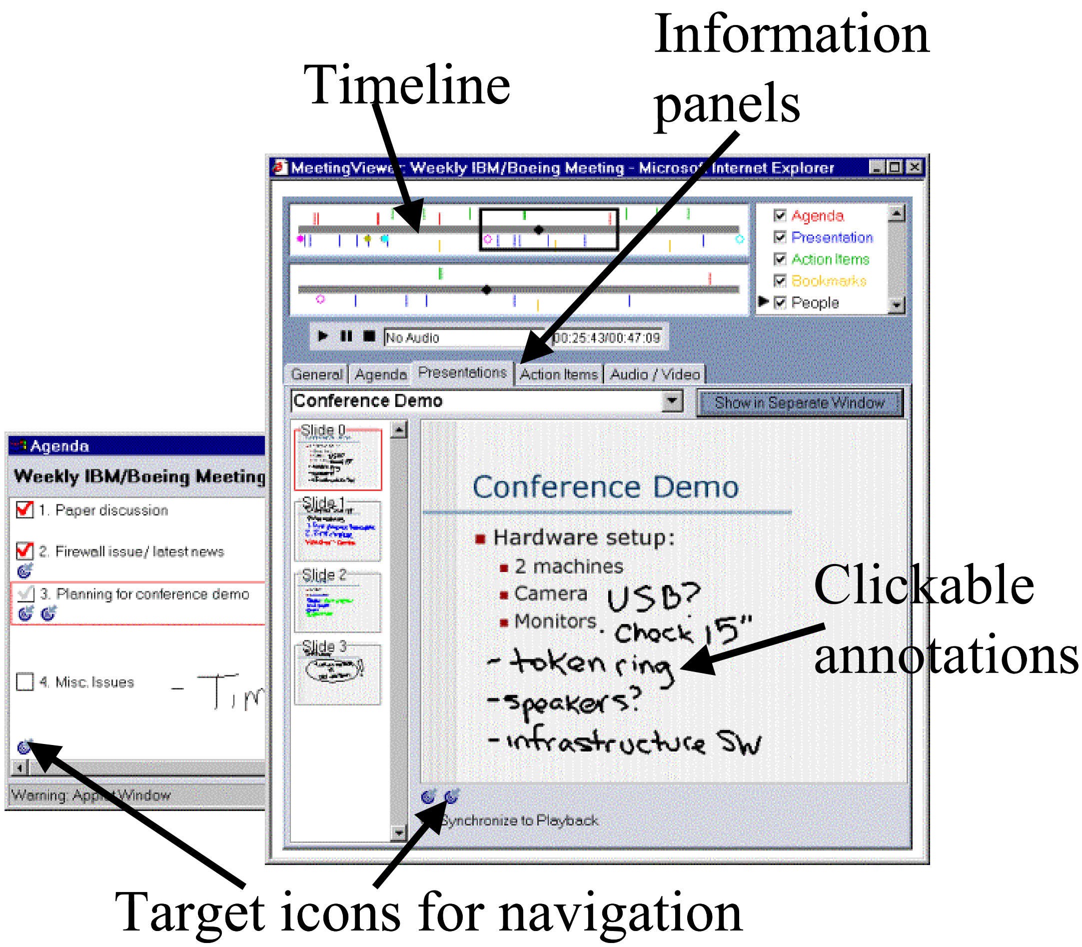

Figure 4.5 Screenshot of MeetingViewer

when that item was discussed. The presentation view shows one presentation at a time. Users can view a slide by clicking on its thumbnail in the thumbnail panel on the left and can use the target icons underneath the slide to go to when that slide was shown in the meeting. Users can also click on slide annotations, both ink and text, to jump to when those annotations were created. Annotations are grayed out if they have not been created at that point in the meeting, but are still clickable.

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 so we can learn more about the types of information users need for various tasks and understand how to design task-oriented views that are simpler and more integrated in the future.

4.4 TeamSpace Implementation

While we refer to the TeamSpace system as a prototype, the system is fully functional and has been deployed at Georgia Tech for several years. With multiple components, some web-based and some collaborative and distributed, that all operate on the same underlying objects, the system is quite complex with components in different languages. The system is very scalable, supporting any number of projects, users, and objects. We faced numerous issues in implementing a capture system for inter-company distributed meetings, such as complicated communication through firewalls that normally is not an issue in academic settings (Fuchs, 2001).

The basic architecture of the TeamSpace system is shown in Figure 4.6. The TeamSpace server consists of servlets and Java Server Pages (JSP) that access and store the data in a commercial database using Enterprise Java Beans (EJB). The main web interface requires the use of Internet Explorer due to the difficulties in supporting multiple web browsers with slightly differing implantations of Java and HTML functionality. Internet Explorer was chosen as it is the supported browser at Boeing. The MeetingClient is a Java application that connects via our own protocol directly to a Meeting Server, also a Java application, which is in charge of distributing messages to the clients. The Meeting Server is also a passive client that stores a current version of the state so that it can update late-joining clients and store all the data permanently on the

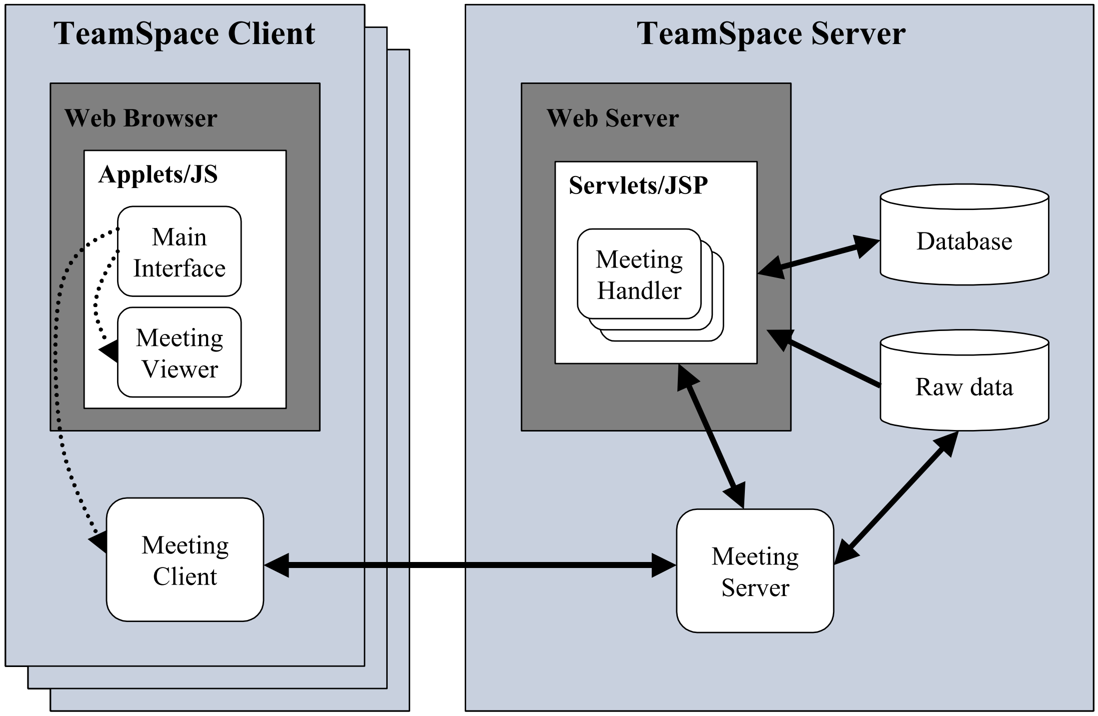

Figure 4.6. The TeamSpace Client/Server architecture.

server. MeetingServer supports and synchronizes any number of clients in any number of meetings. While we use our system to distribute video images, to guarantee audio quality of service we are not digitally transmitting and mixing audio. We expect distributed 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 audio is sent from the client after the recording is completed. The Meeting Server then 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 the database. The MeetingViewer is a Java applet that gathers data from the server using servlets, and communicates through JavaScript to the RealPlayer® browser plugin to control the playback of the audio.

The current server configuration at Georgia Tech consists of an Apache Web

Server, JRun Servlet Engine, and an Oracle database. The server also requires access to an SMTP server in order to email event notices and meeting invitations. The clients need to use Internet Explorer and have Java installed. Capturing requires installation of RealProducer to record audio and our MeetingClient software, which is easily installed with one batch file. MeetingViewer requires the client computer to have RealPlayer and our custom external Java library, as web browsers did not support Swing at the time. This library requirement is no longer necessary and has recently been removed. Finally, MeetingServer must be running on the server for meeting capture to occur.

The entire system was more difficult to implement than we expected. 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 that implemented all aspects of sending and synchronizing data. However, 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, and information is passed without encryption. Additionally, we faced implementation issues that are common to any distributed, collaborative system, such as synchronizing information between any number of clients in a timely manner and handling network outages or other communication problems.

4.5 Evaluation: Access Behavior

One of the goals of this thesis is to understand how users can browse and search captured material. The difficulty with examining this is that real review may be a fairly rare occurrence, and difficult to observe. Thus, we need to use a more controlled study that allows us to observe and question users searching for information. The questions that guided this study were:

- What information or artifacts do users review?
- What role does the audio, presentation, agenda, and action items play in review? How are people using, or how do they want to use, the audio?
- How do people search for static information? How do they search for information within the audio? What indices are they using? What other context or memories are they using? What are the patterns of navigation and replay?
- What are user impressions of meeting review? How much effort are they willing to give toward finding information? What do they like and what do they wish to change?

Teamspace has been used to record the weekly Ubicomp group meetings at Georgia Tech for the past several years. We have used several of these recordings to study access behavior with Meeting Viewer. Ubicomp group members were asked questions regarding previous meetings, and observed using MeetingViewer to find the answers. The results provide insight into the usability of MeetingViewer, how users browse and search for recorded information, and how they make use of the various indices and audio.

4.5.1 Study set up

In this study, members of the Ubicomp group were given several questions regarding a previous meeting. Most of the subjects had attended the meeting, although several had not, or had missed a portion of the meeting. The subjects were instructed to answer the questions as best they could, using MeetingViewer only if desired. Subjects did occasionally remember the answers to questions, but most often used MeetingViewer to find the answers. Subjects were asked to think aloud, and were video-taped, to help determine what internal thoughts, memories, and strategies they used to find information. All interaction with MeetingViewer was logged. After each review session, users were given a questionnaire asking them their impressions on the effectiveness of the searches, and the usefulness of the different artifacts and aspects of TeamSpace.

The questions that subjects answered about the meetings varied greatly, dependent on the content of the meeting. While the questions were certainly not important to all subjects, we tried to represent realistic queries from the meeting. Most questions could not be fully answered without the audio, and were likely not noted during the meeting by anyone as few members of the group take personal notes. At least one question in each session asked for detailed facts, such as a date or name. For example, "What was Gregory's thesis topic?" or "What is the new name of the Yamacraw building?" The answer to this type of question was usually contained only within the audio, often within a segment that was several seconds long and required the user to find that exact audio segment. However, the correct answer was generally clear and unambiguous. Other questions asked for the details of a longer comment or statement made during the meeting. For example, "What house rooms were revealed as significant by Andy's study and why are they significant?" and "What is virtual rear projection?" Answer to these questions were sometimes partially contained within the notes or presentation. The related audio segments were also longer, and users could often find information within or across several segments of the meeting. Answers to these questions were not always as clear, and there could be several "correct" answers. Finally, other questions were more open ended, asking about issues that were discussed, or the resolution or outcomes of a discussion. These questions were covered in a longer discussion, with no one segment containing the "answer." The questions were also more ambiguous, sometimes having multiple possible answers. The notes or presentation usually contained partial answers to such questions. For example, "What are Gregory's concerns with the aware home?" and "What is the overall recommendation for reshaping Jim's paper?" Questions were created by re-listening to the meeting and brainstorming a list of possible questions. Three or four questions were then chosen, attempting to balance the type of question, the topic of the question, and the ways to access the answers.

The study was conducted in two parts. Part one was performed in the fall of 2002 and early spring of 2003, using 7 meetings and 12 subjects, with 14 total sessions. An additional 3 sessions were pilot studies, and were not video-recorded. Only questionnaire responses from these pilot studies are reported in the results. Nine sessions were performed one week or less after the meeting was recorded and five within a month of the meeting. Part two of this study was performed in the fall of 2004, using three meetings, 9 subjects and 18 sessions. This was performed so that each meeting would be used with more subjects, allowing for better comparison of behaviors. Two of the meetings were reviewed less than a week afterwards. The third meeting used was a retreat – a four-hour meeting held at the beginning of the fall of 2004. The review sessions were held three months later. Overall, we evaluated 19 subjects in 32 sessions and 3 pilot sessions.

With the exception of the retreat, all of the meetings lasted between 80 and 100 minutes, typical of the weekly group meetings. All of the meetings had agendas, ranging from 2 to 6 items. Four meetings included one action item. Four meetings included presentations, with two of those containing almost no additional notes or annotations. The remainder of the meetings had one to three pages of notes. There were an average 16.8 events captured for each meeting (not counting the participants joining and leaving) and an additional 21.2 text or ink annotations. Summaries of each meeting are showing in Table 4.1.

Table 4.1. Summary of meeting content. Listed are meeting length in hours and minutes, the number of agenda items, action items, slides, and annotations, as well as the number of events captured.

Meeting #	Meeting Length	Agenda Items	Action Items	Slides (+ notes)	Annotations	Events
1	1:21	3	1	2	19	10
2	1:37	3	0	13	11	25
3	1:09	5	0	1	10	8
4	1:31	3	0	19	10	28
5	1:20	3	0	11	0	20
6	1:10	5	1	3	45	18
7	1:32	4	1	3	81	12
8	1:38	2	0	25	2	30
9	1:14	4	0	1	4	7
10	4:03	6	1	2	30	10
Average	1:40	3.8	0.4	8	21.2	16.8

4.5.2 Results

Participants remembered surprisingly few details from a meeting. Out of 117 total questions, subjects answered 18, or 15%, completely from memory. Several of these questions were easily answered as the topic was about the subject. Of those 18, four questions were answered incorrectly (4/18 = 22%). For all of the remaining questions, subjects attempted to use MeetingViewer to answer the questions. All subjects used MeetingViewer to help answer at least two of their questions in each session, and were not always able to find an answer. Subjects gave up their search and did not answer eight of these questions (8/99 = 8%).

Task performance and behavior varied widely, both because of the different questions and meetings, and individual differences in what participants remembered and how they searched for answers. Results for each session are shown in Table 4.2. Users spent anywhere from 20 seconds to 15 minutes using MeetingViewer to answer a question, averaging 4 minutes with a median of about 3 minutes. They played an average of 11:05 minutes of audio per session, using the timeline an average 43.5 times, and the meeting artifacts an average 5.6 times to navigate the recording. Subjects gave up on a question after anywhere from 1.5 to 12 minutes, averaging 5.5 minutes. Several gave up only after many minutes of searching without success, but others gave up quickly because they had no idea where to look and did not wish to waste time searching. The amount of time a subject was willing to devote to a question varied by person. Three users (3, 12, and 13), each of whom participated in more than one session, seemed to have less patience than the rest of the group, usually answering very quickly. These three accounted for all of the wrong answers from memory, and half of the answers given up. Yet even this differed over sessions. Subject 12 spent a significant amount of time during his first session, possibly because he was unfamiliar with the interface and task. During his final session, Subject 3 uncharacteristically spent 15 minutes on one question, and 10 on another, skipping through audio when other subjects answered the questions much more quickly.

It is difficult to characterize and understand the behavior of the participants with these simple statistics. Participants answered questions with different completeness or depth, some taking more time to find additional details. Subjects fluidly answered the

Table 4.2. Summary of session performance. Times are shown in mm:ss. Questions answered from memory show a time of 0. Times in bold font are wrong or not completed answers. The NA times are not available due to the subject's think-aloud which made determining these times impossible.

		Time spent answering question				Time playing audio	Timeline seeks	Artifact seeks
Meeting #	Subject #	Q1	Q2	Q3	Q4			
1	1	10:10	**12:16**	5:33		25:40	40	9
2	2	1:44	3:19	3:00	**4:59**	13:57	44	0
2	3	4:20	2:54	5:18	0	10:22	35	3
3	4	0:45	2:43	2:36	0:59	10:41	2	3
3	5	2:23	0	0	2:59	3:11	8	2
4	6	6:13	7:53	2:53	12:16	14:36	9	7
5	7	4:16	3:14	0:27		4:14	21	0
5	8	8:01	4:41	1:33		9:51	2	3
6	9	4:45	0	0:59	5:59	12:27	53	3
6	10	3:22	0	1:20	2:00	6:56	15	5
6	11	0	2:40	1:33	2:12	6:49	19	10
7	3	3:04	1:35	5:52	2:36	14:13	10	11
7	6	0	1:03	0	0:29	0:00	0	0
7	12	NA	NA	**NA**	NA	23:15	98	5
8	3	2:00	0:00	1:04	1:13	1:20	8	0
8	13	2:36	2:13	2:41	**2:21**	9:03	81	0
8	14	1:15	4:20	0:00	2:36	3:28	15	3
9	3	**0**	5:19	0	2:24	8:26	50	4
9	6	7:57	0:52	2:13	2:30	12:17	48	1
9	12	**0**	**5:00**	0	**0**	3:11	33	3
9	14	7:52	0:49	1:22	0:19	9:32	56	0
9	15	0	7:42	0	0:53	6:51	36	0
9	16	**1:22**	**7:04**	2:24	1:07	10:31	38	3
9	17	5:56	6:56	0:56	1:03	11:50	160	4
10	3	15:07	11:34	4:26		31:08	97	14
10	6	6:57	5:10	2:37		19:54	91	4
10	12	6:06	**0**	3:38		10:14	45	14
10	13	5:53	**5:15**	3:06		12:25	125	5
10	14	3:59	5:14	3:50		7:27	13	16
10	15	9:04	5:03	1:54		13:17	31	14
10	17	6:40	7:32	5:06		17:46	94	18
10	18	0	6:57	7:24		10:13	15	14
Average				Over all questions:	4:04	11:05	43.5	5.6
St. Dev.					2:58	6:45	39.2	5.4

questions, finding answers to one while looking for another, some taking the time to write in complete sentences, others with as few words as possible. The ways which subjects searched often differed. Some missed opportunities to use indices that would have made searching easier. Others randomly searched and quickly got lucky, while others unfortunately jumped right over the desired audio segment. We further analyzed the logs, think-aloud and questionnaires, looking for patterns of behavior, specific use of indices, and usability issues with MeetingViewer.

4.5.2.1 Behavioral Patterns

In order to gain a better understanding of the results and user behavior for each review session, we created a visualization of each subject's behavior over time, showing where in the meeting recording she was playing, and when and how she navigated the playback. An example of this visualization is shown in Figure 4.7. The x-axis represents

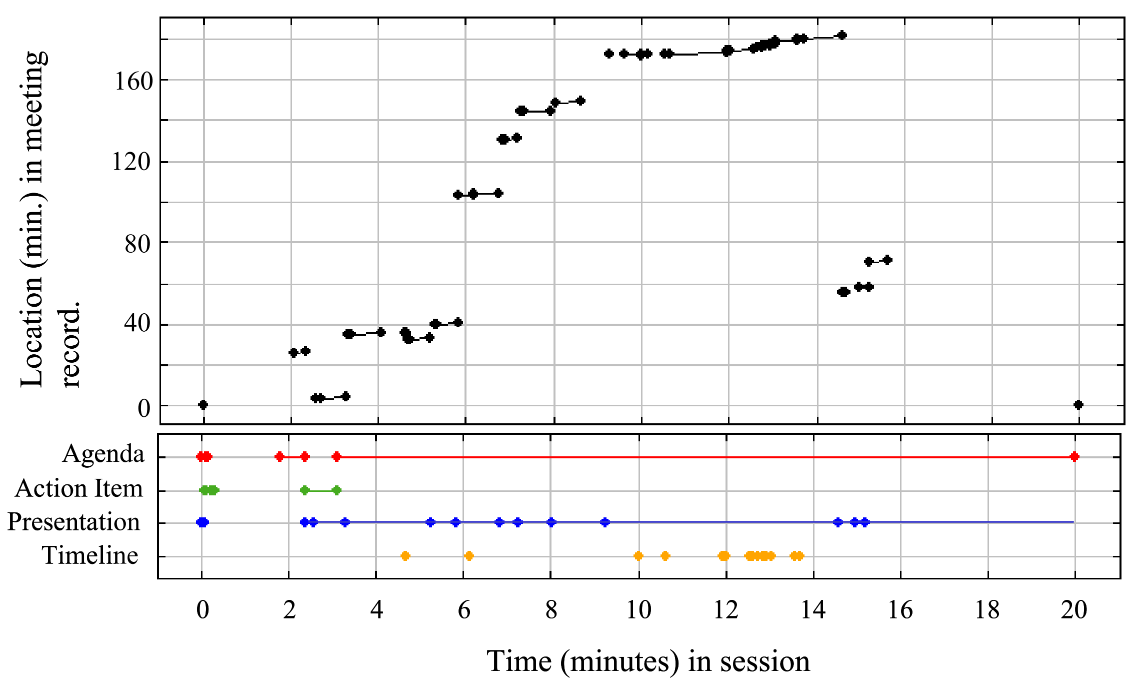

Figure 4.7. Review session for Subject 18.

time, in minutes, elapsed in the review session. In the upper portion of each graph, the y-axis is the location of playback within the meeting record. Black dots denote when the audio was started, stopped, or moved. Lines indicate that audio was playing. The lower portion of the graph indicates what information was being viewed in MeetingViewer and what was used to navigate. In this case, the y-axis is divided into information category. A line for a category indicates that the information panel was being viewed. For example, in Figure 4.7, the user was viewing both the agenda and the presentation for most of his review session. The timeline is always visible, so no line was drawn to indicate this. Dots indicate when an information panel was opened or closed, or when information on that panel was used to move the playback point in the meeting. In Figure 4.7, the presentation was used to navigate the meeting, as well as the timeline.

With these visualizations, we were able to look for and identify patterns and differences in meeting navigation. Figure 4.7 shows Subject 18's review session for Meeting 10, while Figure 4.8 shows Subject 13 reviewing the same meeting, answering the same set of questions. Both review sessions lasted roughly the same amount of time. Yet their review behavior was very different. As the visualizations show, Subject 18 frequently used the presentation, annotations in this case, to move through the meeting. Subject 13 on the other hand preferred the timeline, and mainly navigated in this way.

In matching these visualizations with observations of the sessions, we have identified a number of repeated patterns of behavior. Identifying these behaviors allows us to both characterize user navigation in our study and investigate how the interface supported those behaviors. In turn, this can provide direction to future access interfaces on the behaviors and interface features they may want to provide. In the following

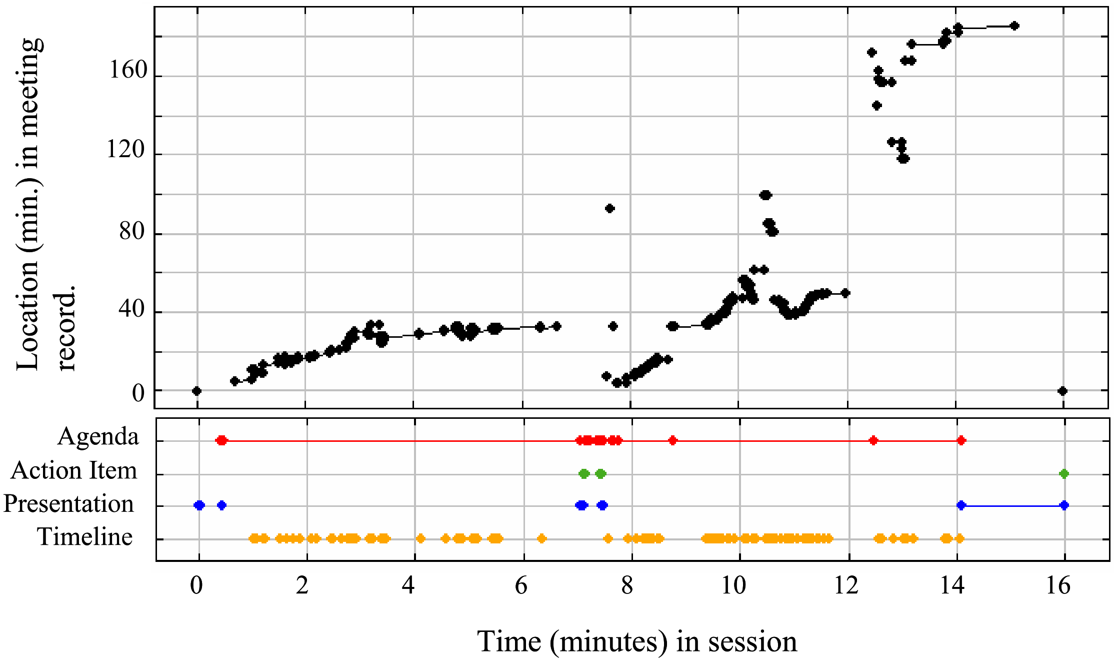

Figure 4.8. Review session for Subject 13.

sections, we define each pattern and demonstrate its use by a user. We also discuss usability issues that occurred and the interface implications for each behavior. We will continue to use portions of Figures 4.7 and 4.8 to illustrate each of the behaviors.

4.5.2.1.1 Scan

Scan involves quickly glancing at the information in the interface to get a feel for what is available. In MeetingViewer this was performed in two ways. Users flipped through all of the information panels very quickly, spending only a few seconds on each. We demonstrate this in Figure 4.9, where subject 18 quickly visited each of the information panels in succession. This example is taken from the beginning of the session of Figure 4.8, where subject 18 exhibited this behavior twice at the beginning of his session, before he began any review. In Figure 4.8, subject 13 also performed a scan at approximately 7 minutes into his review session, flipping through each panel multiple times. When a presentation was given, users similarly flipped through many or all of the slides using the slide thumbnails, again spending only a brief amount of time on each slide.

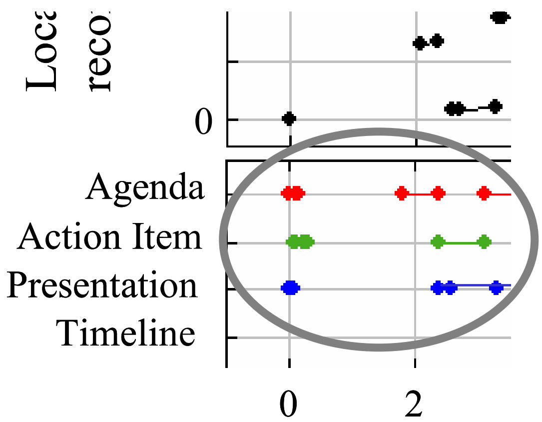

Figure 4.9. Scan behavior.

Usually at least once during the review session, averaging 1.4 times per session, users would scan to get a feel for the information in the interface. This scan usually included the empty Audio/Video panel as well. Some users scanned at the beginning of their session. However, scanning often occurred when the user was lost, frustrated, or waiting for the audio to load.

Scanning was not a difficult task. The information panels were successful in that they were easy to understand and easy to extend. However, knowing that information even existed required going to that panel. On multiple occasions, users would forget to check a particular panel, or forget what they had seen there. They thus missed using an index for quick access, and instead had to search the audio to find an answer. In other words, perhaps we need to eliminate the need to scan multiple panels of information just to see what is available. Additionally, while panels could be detached and viewed at the same time to eliminate the need to scan again, only two subjects ever regularly did this. Another indicated that he wanted this functionality, not realizing it was already available. There will always be a tension between putting too much information on one page, and spreading out the information between panels. In this case, the sparseness of the information would have allowed us to group more information together more effectively.

4.5.2.1.2 Jump

Jump is using an artifact as an index to move directly to a particular point in the meeting. To reiterate, the artifacts in MeetingViewer are agenda items, slides with annotations, and action items. This information provided a semantic structure to the recording and related to the content of the discussion. In Figure 4.10, Subject 18 uses the presentation, specifically annotations on a slide, to jump multiple times through the meeting. This is indicated by the dots on the presentation line, which correspond to jumps in meeting location in the upper portion of the graph. This navigation allowed him to largely ignore most of the meeting and only focus on the small segments he wished to hear. We reported jumps as artifact seeks in the results in Table 4.2. On average, users jumped 5.6 times during a review session, but with much variation. Users often started an information search with one or two jumps to find a segment of interest.

We designed TeamSpace to support jumping and were successful in making this form of navigation relatively easy. However, jumping was not immediately apparent on the interface. As shown in the MeetingViewer interface in Figure 4.4, the user needs to click on an icon that resembles a target underneath the agenda items, action items, and slides in order to jump to that location. Users frequently clicked on the agenda item itself, which did not perform any action. Most eventually hit the icon, although did not always

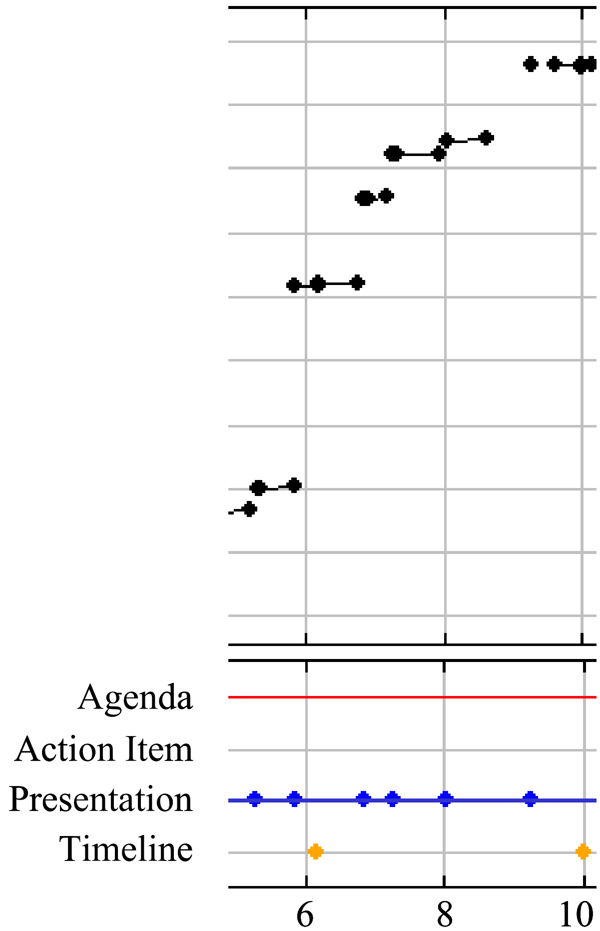

Figure 4.10. Jump behavior. Subject 18 uses annotations to navigate the meeting.

realize that this was what was successful. This was particularly apparent in Meeting 10. One particular agenda item did not get selected during the capture of the meeting, and thus, no icon appeared for that item in MeetingViewer, even though everyone was aware that they had in fact discussed the topic. Most users clicked multiple times on and around the agenda item, wondering how in the world to jump to that point. The target icon was even more problematic for the slides. Even if the user understood the target icon functionality on the agenda, he did not always realize that same functionality was available for a slide, or just did not notice the small icon. Thus, while annotations on a slide were frequently used to jump, slides were infrequently used.

Using artifacts to allow users to jump to a particular point was easy to understand. When successful, jumping required very little interaction and was used whenever available. For example, even when the answer to a question was written as an annotation on the slide, subjects frequently clicked on that annotation to jump the playback and listen to the audio while they wrote their answer, even though the audio was not required. This may indicate that audio would be consulted more frequently if doing so was only one click away while viewing other information. However, we need to make it very apparent how to perform the jump action. Our usability issue with the target icon did affect this behavior and confuse users. More importantly, the artifacts themselves did not show the flow of time through them. This was confusing when an agenda item or slide was visited multiple times throughout the meeting. While the timeline did show this information, the lack of labels meant that the events on the timeline could not be matched to the corresponding agenda items or slides, making them less useful. Thus, linking static information such as a note, to the dynamic events will also aid the user and make navigation more clear.

4.5.2.1.3 Skim

Skim is using the timeline to systematically navigate through the meeting and audio. This behavior was used when looking for a particular topic or segment of interest in the audio. Sometimes this behavior was also used to get a general idea of what was discussed throughout a portion of the meeting. This behavior is indicated by using the timeline repeatedly for small navigations in the meeting. For example, in figure 4.11, Subject 13 spends most of the review session skimming through portions of the meeting. Skimming was almost always done forward in time; users would only skim backwards if

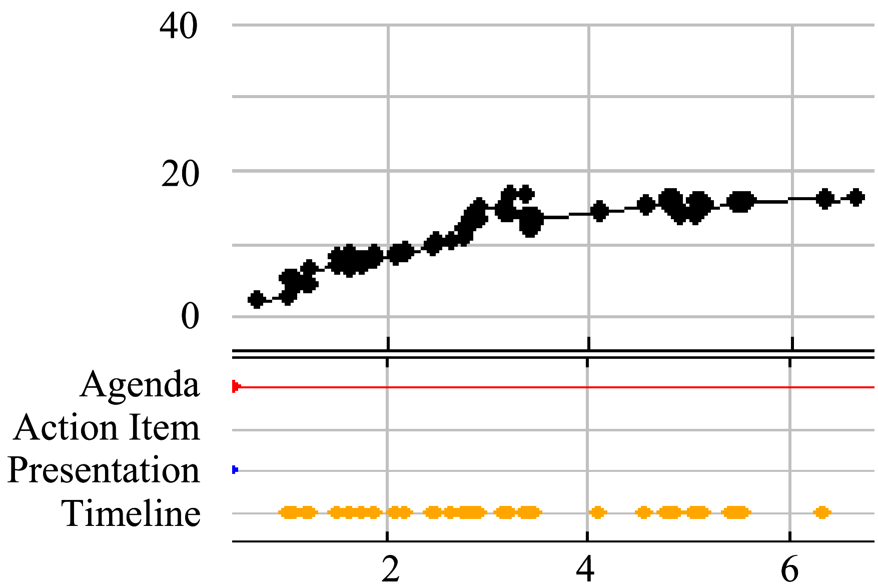

Figure 4.11. Skimming behavior. Subject 13 uses the timeline to move through the meeting.

they realized that they had overshot an area of interest. Users often listened to the audio for a very short amount of time, usually about 1 to 3 seconds, before moving to the next point. The distance of skips varied both by person and by task. Some preferred smaller moves, about 10 seconds. Others skipped a minute at a time. A few participants used the event lines of the timeline to help them know where they might want to move, but most appeared to rely simply on time to help them know where they were.

Skimming could be very time consuming, especially if the area of interest was large. For example, for Meeting 9 in Table 4.2, questions 1 and 2 required some amount of skimming to find, and users who searched spent an average 5:09 to answer those questions. Users spent much longer searching for answers to these questions than they did for questions 3 and 4, where time spent averaged only 1:22. Users could spend more than 10 minutes skimming just to answer one question. However, this task was generally not frustrating as long as the user was making progress. While most people thought they had a good idea of the order of topics, they were often wrong, causing them to skip over a part of interest, or be looking in completely the wrong area. If this occurred, a user rarely wanted to re-skim that portion of the meeting and often gave up. Certain meetings seemed to be easier to skim than others, based upon the content and language of the discussion. The speaker's identity and the words used often gave a very quick indication of the topic. However, meetings with lots of informal comments, chitchat, long silences, and tangents seemed to be more difficult to skim. Users had to listen for longer to determine where they were in the meeting before skimming again. Users had different thresholds for time spent skimming. Some were more willing than others to spend time searching through a meeting before giving up, and even the same user could have different thresholds on different days. An additional confound was that if skimming was making progress, users sometimes did not check for another kind of index first, making their search longer and more difficult.

The timeline was extremely flexible; users could move any amount that they wished. Yet, this flexibility was often not necessary, and instead, introduced confusion. The user was required to judge the distance of their move based on the physical distance on the timeline, and the result of a click was not previewed ahead of time. This seemed to be more problematic the smaller the move. Participants would use either of the two timelines depending on how big a skip they desired. It was easier to make finer skips using the lower timeline, however it could be difficult to move the focus box on the upper timeline due to the small pixel size of the box. Several users requested a fast forward button that moved ahead a pre-set amount of time, in order to facilitate skimming.

Several other users requested ways to automatically skip useless audio, such as dead space or chitchat. This could sometimes decrease the amount of audio the user listens to before moving on, potentially improving skimming behavior.

As a final note, the frequent skimming we observed would not have been possible if moving of any sort resulted in an audio streaming delay. Given that users would sometimes only listen to audio for one or two seconds before moving on, even one second of delay would have interrupted their behavior. Two of the subjects who have been frequent users of educational capture systems commented on their appreciation of no audio delay.

4.5.2.1.4 Honing

Honing could be considered a form of skimming, and the interaction is the same: using the timeline to skip through audio to find a particular segment. However, honing is more fine-grained, with a more focused goal. The general segment of interest has been discovered; the user basically knows where he is in the meeting. He just needs to find the exact point he is looking for. In this case, the user is skipping smaller amounts, usually 5 to 20 seconds of the audio, sometimes as little as a second or two. We show an example of honing in Figure 4.12, which looks similar to skimming, but with smaller movements.

Users frequently used the lower timeline for this behavior, and had difficulty if they did not. This was especially apparent in the retreat, where the upper timeline represented four hours. Thus, moving 10 seconds on this timeline was almost impossible. One difficulty is that the use of the lower timeline was not immediately obvious to everyone. Also, since the focus box was difficult to move due to its small size, an initial attempt sometimes failed and led the user to not try again.

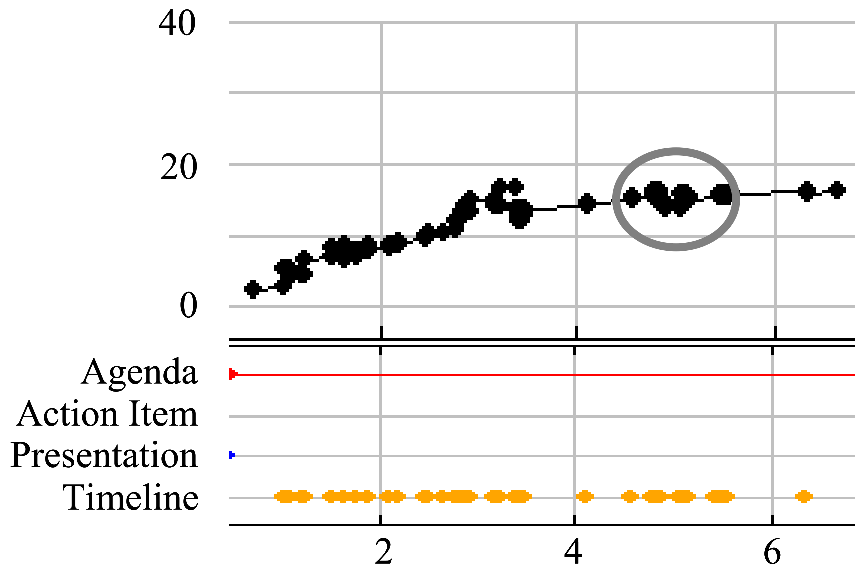

Figure 4.12. Honing behavior. Highlighted above, subject 13 looks for the exact statement.

We separate honing from skimming because the occurrence and implications for the two behaviors may be different. Honing occurs less often than skimming, and only when users are trying to find an exact piece of audio. A fast forward button would improve both behaviors, but the size of the desired move may be different. Yet, while honing may occur less often than skimming in this study, we observed that users were generally impatient and often moved ahead just a few seconds rather than listen to that audio to get to their desired point. As in skimming, this behavior was only possible because there was no audio delay and the timeline allowed these very small movements.

4.5.2.1.5 Replay

Replay is moving backward and playing the same segment of the meeting record another time. An example is shown in Figure 4.13, where the user moves back a very small amount and listens to the same segment of audio 3 times. Users frequently replayed the same small segment of audio multiple times to hear an exact statement. This sometimes occurred three or four times if the audio segment was difficult to hear or if the subject was exactly quoting a phrase as an answer. This behavior was generally performed using the timeline. However, if the audio segment started at a particular index, that index was used instead.

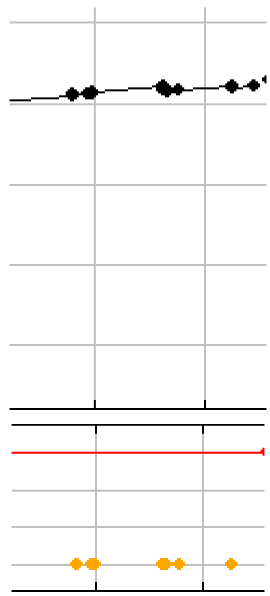

Figure 4.13. Replay behavior. From Subject 17 in Meeting 10. The subject replays the same meeting segment 3 times.

As with skimming and honing, the use of the timeline could introduce problems. The user had to determine how far back she wanted to move and the corresponding timeline location. Sometimes users did not skip back far enough. Sometimes they went too far and either had to hone again or listen to more audio than needed. Similar to improvements in skimming, users requested a reverse button to move back a pre-set amount to aid in replay.

4.5.2.1.6 Random

Random is a move, using the timeline, with no particular target. This is often indicated by several fairly large moves with no apparent pattern of navigation. Figure 4.14 shows Subject 13 do several random moves, followed by the subject listening to the audio for awhile. Subjects stated that when they had no idea where to look, they sometimes randomly moved about on the timeline. One subject even said she used a binary search approach to narrow down her search area on the timeline. This behavior would usually not last long, as a subject would frequently find some segment of interest within a few clicks and begin skimming or honing. Randomly moving was also remarkably successful in finding a particular piece of audio. A subject did occasionally happen across an answer without really knowing where she was looking.

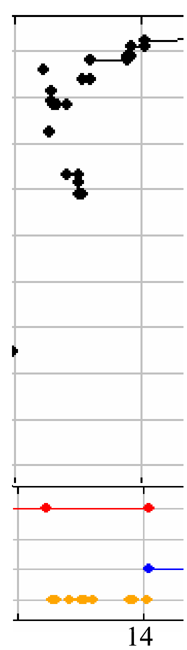

Figure 4.14. Random behavior. Subject 13 makes several random moves.

4.5.2.1.7 Order of behaviors

While the order of these behaviors was greatly dependent on the user and the question, a prototypical, successful, search for information was as follows. The subject would start his use of MeetingViewer by looking at the agenda. If he saw an agenda item that corresponded to the topic he was looking for, he jumped to the beginning of that agenda item. If that agenda topic covered a long period of time, the subject would skim through the audio to find the more detailed topic he desired. Then the user would hone in on the exact answer and replay the answer once found. At some point, the subject scanned the information panels. He used an annotation on a slide to jump to a topic, and again skimmed and honed until the desired location was reached. In looking at our two examples, we see that subject 18 performed less skimming than usual while subject 13 performed more. A more balanced session can be seen in Figure 4.15, where subject 11

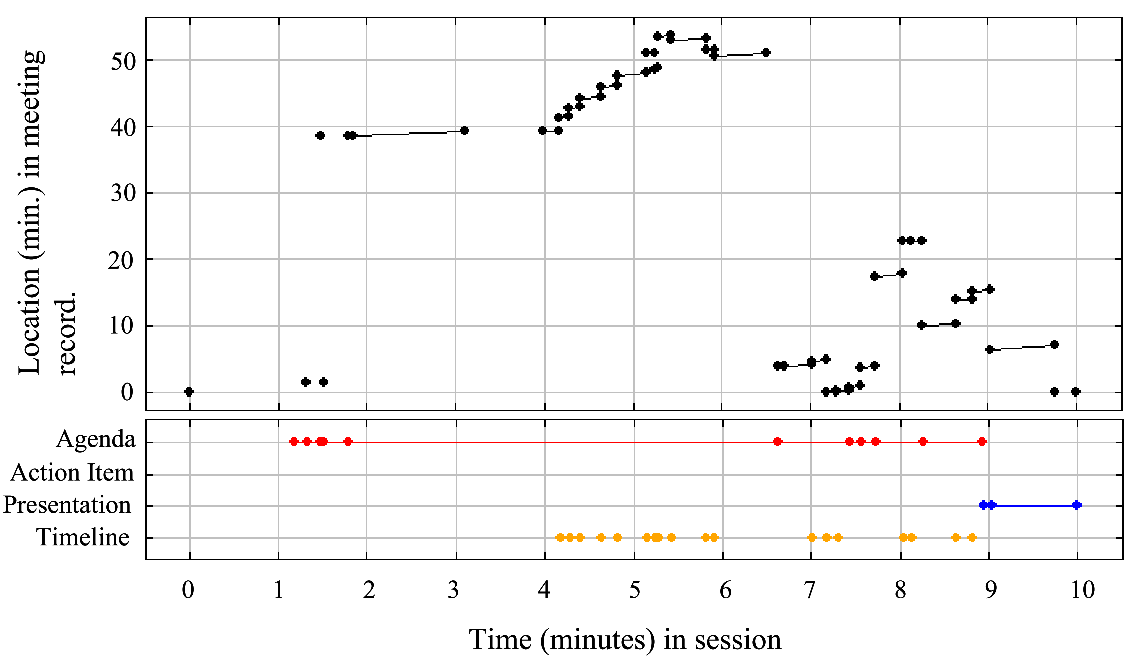

Figure 4.15. Prototypical behavior. Subject 11 in Meeting 6.

shows most of this prototypical sequence, as she initially uses the agenda to find an area of interest, then uses the timeline to skim further. She finally visited the presentation towards the end of her session and used it to jump one final time.

4.5.2.2 Indices

The most important measure of an access interface is how well it supports users in finding the information they want. And, as we have emphasized, the captured artifacts and indices are critical to supporting the browsing and searching for information. On the post-task questionnaire, subjects were asked, "How easy or difficult was it to find the information you were looking for?" The results of this question are shown in Table 4.3. This answer was rated "Moderately easy" in 20 questionnaires, "Not easy or difficult" in 8, and "Moderately difficult" in 7. Using a 5-point scale, this led to an average of 3.4, somewhere between "Moderately easy" and "Not easy or difficult." Many of the difficult ratings were given when users spent a significant portion of their time skimming the interface. Poorer scores were also given more often for sessions that required some amount of skimming.

Table 4.3. Responses to "How easy or difficult was it to find the information you were looking for?"

	Very easy (5)	Moderately easy (4)	Not easy or difficult (3)	Moderately difficult (2)	Very difficult (1)
# of Responses	0	20	8	7	0

Users clearly find information faster when they are able to use indices to find areas of interest, or even the exact desired location. The more indices we can provide, the more likely we are to improve users' searches for information. Every index we provided was used, including the rare action item. Over all sessions, the agenda was used to navigate 78 times, action items 3 times, slides 8 times, and annotations on slides were used 188 times. Indices that provide semantic structure of the recording were consulted first. The agenda was almost always viewed and used to start navigation. In the questionnaires, five subjects stated they wanted more agenda items, or more fine-grained topic changes, to provide additional structure. The more time between the activity and the review, the more likely users will need general, high-level structures to serve as an overview and reminder of what occurred in the meeting. This structure will in turn, spark other memories of the content they are looking for and its context.

While high-level structure is useful for getting started, users desire other artifacts or indices that are directly connected to the content of the meeting. For example, the presentations and annotations contained useful content, and also structured the meeting record. On the questionnaire, we asked the subjects to rank order the indices and audio based on their usefulness to their task, not ranking any items that were not useful. 28 out of the 35 responses ranked the agenda and presentation as useful. Overall, the agenda and the presentation were rated about the same usefulness. In comparison, the people join and leave events were ranked useful in only 12 responses, and generally were ranked lower than the agenda and presentation. In feedback about what would facilitate their tasks, four participants wanted more notes to use, and 4 others requested just more indices in general. Overall, 11 subjects reported wanting additional indices in some form to help find information. In other words, subjects seemed to value any of the indices that related to the content of the meeting, and seemed to want as much of this as possible.

This study also shows that any type of index can and will help users find details within the recorded meeting. Anything that can be accurately captured is likely to be useful to someone at some point. Users did make use of the join and leave events, using the knowledge that what they were looking for occurred after a particular person entered or exited the meeting. One of the most requested improvements to MeetingViewer was various speaker context. Four subjects suggested speaker identification and four others, speaker change information. Several others also wanted detection of silence and chitchat. Not only can this speaker context help users search based on who was speaking, they can also indicate the type of discussion occurring – such as a presentation with one speaker, versus a discussion with rapid switches between speakers. Speaker changes can also indicate topic changes during certain types of meetings, as different speakers address different topics. When the content-based indices have narrowed down the area as much as possible, these indices would then speed the process of skimming and honing on the desired details. These and other forms of context-based indices are being actively researched in multimedia and information retrieval communities.

The other heavily requested addition was a searchable transcript or use of keyword spotting. This would allow keywords to function as a content-based index, narrowing down the areas of interest. A transcript could also eliminate the need for audio skimming, substituting text skimming instead, which would likely be faster. In this case, audio would only need to be played when the transcript was not sufficient to convey what was said, either because of recognition errors or because other verbal cues are not present. As speech-to-text technology improves, we will be able to take advantage of these capabilities to improve access to meeting content. However, as we will demonstrate in the TagViewer use in the next chapter, we do not believe that a transcript will eliminate the need for the content-based indices that structure the discussion.

In TeamSpace, we did not explore post-hoc or review-oriented indices, i.e. indices that were created after the meeting based on users' review of the recording. Several users also expressed a desire to bookmark areas of interest. However, this would only be useful if the user, or perhaps others, were going to later return to that portion of the meeting, which did not occur in this study. Users may also benefit with knowing where they have been in the recording. While not common, a few users did become lost and unintentionally revisit the same portion of the meeting multiple times. Indications of where the user has visited may lower these occurrences, and also help with replay. For example, if the user skips over a desired segment while skimming, he could focus on only the areas he had not visited yet.

One issue with any of these indices is that almost none of the users ever interacted with the capture application. This led to misunderstandings of what certain indices meant and how they were created. For example, no one understood what the bookmark events were supposed to represent since we rarely used them and no one was aware that the functionality even existed to capture them. Also, users wanted the people events to represent speaker identification, which our system could not do, instead of the join and leave events captured by checking off participants in the capture tool. When a problem occurred, such as an agenda item that accidentally did not get selected during the meeting, users attributed the problem to the access interface instead of a capture issue. Improved labeling or a help interface could alleviate some of the confusion. However, conveying the meanings and potential problems with various indices will always be a challenge when the reviewer is not familiar with the capture capabilities and how the events are generated.

4.5.2.3 Access Behavior Study Conclusion

This evaluation also allowed us to observe a number of common behaviors in accessing meeting information. The study confirms our desire to provide a number of indices based on the artifacts in the meeting. These content-related indices were critical to quickly finding a portion of the meeting, and users wanted more of them. In other words, applications should support indices for jumping, and support skimming explicitly when necessary. Subjects would further benefit from any index or feature that allows them to quickly search through the audio, including fast forward and reverse buttons, speaker changes, and speeding up audio playback.

Despite the decade and a half of meeting capture research, this is the most detailed evaluation of the behaviors of an access interface in this domain that we are aware of. Even in other domains, only the evaluation of eClass (Brotherton, 2004) has provided comparable details, yet still did not discuss the navigation behavior at this level of abstraction. Instead, the results focused on entire sessions and whether users played straight through one segment, replayed a segment, or skipped around. Thus, our study has revealed even more details and specific interface issues that relate to those behaviors. However, the task supported in this evaluation was a search task, where users found a desired portion once and never returned. Thus, there are likely to be move behaviors and issues that are important to other tasks. For example, Moran et al. observed a behavior they called salvaging in their evaluation of Tivoli (Moran, 1997), the high level behavior of finding, pulling out, and structuring interesting portions of the meeting record. However, many of these basic browsing and search behaviors are likely to be a part of any access of meeting information, and are thus important to understand and support.

In general, MeetingViewer was usable by the subjects during their task. Most were able to find answers to the questions, and many made very positive comments about the interface. The interesting thing to note about MeetingViewer is that it cannot merely be usable. It must be "walk up and usable." Users were only ever given very brief training the first time they used the interface. And for some, this training was several years ago with only one or two subsequent uses per year. As we further discuss in the following section on the authentic use of TeamSpace, infrequent use means that users will easily forget what they learned and discovered during their last interaction.

While MeetingViewer was generally useable we found several issues that arose because a single user captured the meeting on a laptop instead of conducting the meeting using the capture application on a viewable whiteboard. Users did not experience or view the capture functionality, and thus, did not always fully understand the review capabilities. For example, we used annotations on a blank slide as notes, which did not function well for that purpose. Additionally, users occasionally forgot to look for notes under the "Presentation" panel on MeetingViewer. Something more similar to the Quindi MeetingCompanion product (http://www.quindi.com/) would be preferred for note-taking. These problems demonstrate the challenges in providing users an understanding of the capture capabilities, and the need to design features that support the desired information in the meeting. We additionally found other, more minor usability problems not already discussed that are listed in Appendix B. To summarize, usability is still a big issue, and even small interface improvements would improve a user's ability to search for and find information in the meeting record.

4.6 Evaluation: Motivations of Use

In order to understand the motivation for using capture materials, we must let groups use the technology in their own settings for their own purposes. This means introducing capture services with as little change to their existing meetings as possible, encouraging repeat usage, then observing how people use captured information. In order to gain repeated usage, we can suggest ways to use capture during the meeting, reasons for reviewing the meeting, and encourage users to look at a meeting at least once. In the end, however, users' own motivations will either lead them to use or not use captured services in their current form.

In order to help us gather these motivations at the time of access, we queried users for their reasons for using MeetingViewer in a dialogue box as the interface was loading. The options were "Missed the meeting", "Review the action items", "Review a presentation", "Find detailed information", "Just browsing", and "Other" with a text box for other answers. The user could choose any number of options, or none before continuing. This allowed us to get a quick indication of the user's intentions with minimal effort for the user.

Our original goal in creating TeamSpace was to be able to deploy the system to groups and observe how and why people used captured meeting information. We attempted this deployment with a number of groups with little success. While we did not learn as much about the benefits of meeting capture, our failures still provide lessons on the motivations of these groups and the appropriateness of TeamSpace, or meeting capture in general, for them. The most extensive use of TeamSpace has been within our own research group, the Ubicomp group, a group familiar and comfortable with capture and access technologies. While our use of TeamSpace was aided by the particularly motivated TeamSpace researcher, the system was incorporated into the group's regular practices and had real, naturally motivated uses. In the following sections, we chronologically discuss each deployment attempt and the lessons learned, and end with the use of TeamSpace by the Ubicomp research group.

4.6.1 The TeamSpace team

During design and implementation, the distributed TeamSpace team used the software to conduct and record their meetings as much as possible. The meetings generally consisted of two members in an office at Boeing in Seattle, three to four members in a lab with an electronic whiteboard at IBM Research in Hawthorne, NY, and occasionally one member at her desk at Georgia Tech in Atlanta. We successfully recorded over a dozen weekly meetings, as well as a distributed presentation for a yearly project review, and two 4-hour strategy sessions. The meeting environment was useful for brainstorming and for discussing presentations. Team members revisited several meetings to view the annotations we made, or review a missed meeting.

The meeting dynamics changed in small, but noticeable ways. Prior to the completion of the prototype, the group did not make much use of an agenda, although team members would often privately note items to discuss prior to the meeting. These items became publicly listed in the agenda, yet this was not seen as too important as most meetings only contained a few agenda items. However, the agenda was used extensively and was much appreciated for the 4-hour strategy meetings. Additionally, team members shared information during meetings using PowerPoint slides. As we did not have a standard way to share information previously, this was not a problem, although we would have certainly liked ways to share information in other formats. Finally, the group experienced an interesting phenomenon common to distributed collaborative systems – knowing what the other party is seeing. Group members frequently asked if others could see that someone had joined or made an annotation. While this was often necessary when the prototype was not stable, we continued to ask these questions even after the system was fully deployed.

Our own initial use of the system was encouraging. Captured information was most useful for team members who missed the meeting, but we did find moderate needs to review information in other situations. Note-taking prior to TeamSpace was sparse. Yet, the annotated slides were useful to review brainstorming or edit a presentation. We occasionally discussed an action item and reviewed the list the following week to make sure that we did not forget to do anything. Thus, the group did come to rely on the system to record activities and did find TeamSpace useful. We were all motivated and experienced users, however, and such familiarity clearly aided our use. We designed TeamSpace to be used in this type of distributed situation where MeetingClient helped to conduct our meetings, not just capture them. Our distribution may have aided our access as well, as we could sometimes more easily consult the meeting record than each other. Yet, we were the only distributed team who used TeamSpace, and a number of features that were designed for distributed teams were not as useful for co-located groups.

4.6.2 Boeing

We attempted to deploy TeamSpace to two separate groups in Boeing in 2001. The first was for a group that met weekly to discuss computer architecture technology and issues for a new aircraft. The group members were distributed across work sites, and many often had to miss meetings. This was the type of group we wanted to support with TeamSpace and initial interviews with group members about the upcoming deployment were positive and encouraging. The meetings seemed relatively important where members often wanted to share the discussion with missing group members. Unfortunately, the members of the group were so busy with tight development deadlines, that before we could fully deploy, they disbanded and stopped meeting altogether.

The second group was an administrative team that met weekly to discuss planning, status, and personnel issues. We were able to capture several of their meetings. However, the study ended when it became clear that the group did not have any desire or need to review meetings. An assistant already served as minute-taker. Group members would just consult her if there were any details they wanted to remember. At this point, our collaboration with Boeing ended and we did not explore other groups in the organization. We did perform initial interviews with several members of each group. The first group was particularly interested in capture, and very positive about the possibility of using TeamSpace. The second group was also positive despite their later lack of interest, and only had privacy concerns with certain portions of the meeting where sensitive personnel issues were discussed. Thus, we were encouraged by everyone's impressions of capture, but unfortunately did not have a successful deployment. With this, we moved to studying local groups to make deployment and evaluation easier.

4.6.3 SE Research Group

We deployed TeamSpace to a research scientist in a software engineering research group in the College of Computing at Georgia Tech. The research scientist, who we will refer to as Brian, conducted up to 10 meetings a week with individuals or small groups of students in his office. We thought Brian would be a good candidate for using TeamSpace, as he has careful processes in place to take notes and remember the outcomes of all of his meetings. For several months, Brian used TeamSpace to record many of his weekly meetings with students. Prior to the deployment, we interviewed Brian and gave questionnaires to the students to understand existing practices. Nine students participated. We then followed with another questionnaire for the students and another interview with Brian to understand their use and opinions of TeamSpace. This deployment showed that meeting capture and access was very well received in concept, but the barriers to using TeamSpace are still too great when existing practices suffice. We first describe Brian's existing practices, then his use and feedback on TeamSpace.

4.6.3.1 Existing practice

Brian holds a number of regular, weekly or biweekly meetings with undergraduate and graduate students in his office on a number of research projects. In these meetings, they discuss weekly progress, solve problems that have come up during the week, and decide on new tasks and future directions. They may brainstorm, review presentations or papers, or go over code depending on the needs of the project. Students frequently bring their laptops, and may use it to show something for discussion. Brian takes notes for each meeting, and maintains a file folder for each student to hold all of those notes. He began doing this after noticing that students were not taking sufficient notes, and they occasionally had to revisit an issue or solution because of this. Students occasionally ask for a copy of the notes for themselves. Additionally, each student maintains a virtual directory using CVS. Students are responsible for maintaining an electronic list of to-do items for each week that is placed in that directory. Brian frequently uses the whiteboard in his office to brainstorm and draw diagrams. To store this information, he sometimes takes a picture with a digital camera and stores the picture in the CVS directory.

Brian stated that he often looks at the notes just prior to the meeting. He is mainly concerned with remembering the to-do items, and what issues are still unresolved and need to be discussed. Occasionally, he may look for something in the notes even during a meeting to remember a previous discussion. Thus, the most important things both he and the students want to remember are the action items, and sometimes the details of a description or problem solution. Brian thought that this process could be improved with these notes and to-do lists being automatically created in an electronic format that was then editable. He was extremely positive about the ability to have the details of a discussion recorded so that they are not forgotten. Only one student reported having any difficulty remembering any of these meeting details, however. Thus student interest seemed limited.

4.6.3.2 TeamSpace Use

Brian was initially given TeamSpace to try in 2002. He recorded 8 meetings and five action items, although two meetings appear to simply be tests of the software. One meeting included several posters imported as presentations. This initial use of TeamSpace was promising, and was reported as desirable. However, network problems made the software unreliable. He discontinued his use while we repaired these problems. We then initiated a full study in the spring of 2003, including the interviews and questionnaires. To jumpstart his use, we requested that he record all of his meetings, with permission of the students, for several weeks. After that, he was free to use TeamSpace as he wished.

Brian used his laptop, which sits on his desk, to run the TeamSpace software. Additionally, we gave him a Wacom pen-based LCD tablet to function as a monitor and a way to input ink strokes. He placed the tablet on the side of the table that he and his students met around. Brian was very diligent in his use of TeamSpace, and recorded 26 meetings over two months. Many of these meetings were between 20 and 30 minutes in length. Most meetings contained zero or one agenda item, and several contained one or two action items, for a total of 10 action items. Only one meeting had a presentation, and one other contained one page of notes – a detailed hand-drawn diagram with several typed labels. Many of the meetings had no artifacts or events whatsoever, and were basically just audio recordings.

Despite the high number of captured meetings, there was very little access. And much of this access appears to be by Brian, out of curiosity or ensuring that things were being recorded as he expected. MeetingViewer was used 15 times to review these meetings. There were several other uses reviewing test meetings. Even of the 15, 4 review sessions lasted less than 30 seconds, all in succession. We are unsure whether Brian was searching for a particular meeting, or whether he was testing something in MeetingViewer. Five more review sessions were less than one minute. Brian may have again been ensuring that recording occurred or been demonstrating the viewer to someone. Only in 5 reviews was any real amount of audio replayed. In all but one case, Brian chose that he was “Just browsing” when MeetingViewer was opened. In the other case, he was “finding detailed information.” This was the meeting that contained the diagram, and he was viewing this diagram. He attempted to play audio, but the audio does not appear to have loaded and started playing before he exited MeetingViewer.

Brian believed that a couple of the students reviewed a meeting. Yet while three students logged in to the TeamSpace web interface, only one student viewed one of his meetings, spending several minutes using MeetingViewer. He flipped through the information panels, which were empty for this particular meeting, and played several minutes of audio using the timeline to navigate. He was the only student who reported using TeamSpace. Several students reported that they continued their current practice of taking notes and to-do items on their own. So TeamSpace does not appear to have affected their normal processes, nor improved upon them. Thus, with the exception of the diagram review, we have little evidence that review was useful for Brian or the student.

4.6.3.3 Impressions of Use

In the questionnaires and interview, we queried about the changes to the meeting or concerns with recording. The biggest change students noted was that it took time to set up TeamSpace and create the meeting, causing the student to wait. Brian did not generally create the meeting ahead of time, instead doing it on the fly when the meeting started. This reduced his time in setting up a meeting, but detracted from the student's experience. Several students commented that they did have to watch more carefully what they said. But none reported being very bothered by this. Brian further reflected on this, noting that it was "slightly embarrassing" initially, but quickly everyone seemed to get used to it. He noted that he was occasionally more careful making negative comments about other people. But he did not consider this to be a concern, as those comments are rare and the audio can be muted if necessary. However, Brian would not want these recordings to be publicly available. He would also prefer the recordings only be accessible by those in attendance, but group accessibility was not a big concern. The most surprising comment was that Brian felt that TeamSpace could have a positive effect on the meeting itself, keeping it more on track, as people diverge from the topic at hand less. We did not get an indication that this occurred for Brian though.

Brian was very positive about meeting capture and access and its benefits, although the problems with TeamSpace made this particular prototype not very useful. The action items "are pretty cool" and perhaps the most useful aspect for his students. This is not surprising given that his current practices also involve keeping track of action items. He suggested multiple line action items so more details could be entered (action items currently consist of a one-line title). He did not make great use of the notes because the resolution of ink strokes was not fine enough. But he thought improved ink and OCR might be useful for taking notes, allowing them to be edited and pasted into other artifacts. The notes in their current form are not re-usable. Brian felt that the audio recording was the most important part of this kind of software, because there are already ways to record the other information, but the details of the discussion are also important to remember sometimes. However, his favorite aspect of the experience was not TeamSpace, but was actually the Wacom tablet that enabled group interaction around a monitor. If more of those interactions could be easily captured, then perhaps TeamSpace would have been more useful.

The biggest barrier for Brian was that TeamSpace was not integrated with his existing tools. He works mostly in a Unix environment, and thus has to move to a new environment just to use TeamSpace. Additionally, creating meetings was tedious and was repeated work. The meetings were already in his calendar program, but needed to be re-created with additional information in TeamSpace. TeamSpace also does not provide any repeated meeting capability to decrease this work. The interaction with the web interface was somewhat cumbersome for him overall, as all interactions are initiated from this interface. A quick-start option easily accessed from his desktop would have helped. He was generally pleased with both the MeetingClient and MeetingViewer interfaces, although did not interact with either a great deal.

Thus, despite our goals of integrating capture with useful artifacts, this integration was not nearly enough to make TeamSpace easily useable by Brian and his students. Rather than simply re-creating important objects, the software would need to integrate or share information with already existing software for scheduling, tracking action items, or running programs. TeamSpace required time to set up and use and we feel fortunate that Brian was willing to spend this time as long as he did. The concept, however, was well received. Brian valued remembering information, and was very positive about recording audio and the benefits that may bring. Yet for these meetings, recording the notes and action items in more traditional methods was easier and sufficed. So the potential benefits of audio recording did not outweigh the costs associated with changing the current practice.

4.6.4 Senior Design projects

We decided to try to deploy TeamSpace to student project teams working on senior design projects as a computer science course requirement. Teams of 4 to 5 students work throughout the semester in designing and implementing software for a faculty member customer. As these are newly formed teams, they would have few existing processes and tool support, and may be more open to using the capabilities of TeamSpace. We explained TeamSpace and offered to help set up and run the software to two sections of senior undergraduate students. Only one project group, with five members, volunteered. These students made good use of the document management capabilities of TeamSpace. They maintained 15 documents, including UML diagrams, project plans and design specifications for their project. Most of these documents were downloaded by other members in the group, with later changes and new version being tracked and uploaded. One group member created 15 meetings, yet did not record any of them. They merely served as visible reminders to his other group members. Thus, TeamSpace served the group by providing a nice group calendar and shared file system, yet we could not evaluate the capture and access capabilities. Group members stated that their meetings were rather ad hoc, without agendas. They also felt little need to revisit decisions or discussions for such a time-constrained project. Perhaps if they were more organized, or working on a longer-term project, they would have tried the meeting capture capability.

While this use of TeamSpace was not helpful in evaluating meeting capture and access, it does confirm the usefulness of shared workspaces and thus our decision to integrate capture and access into such a group space. Perhaps additional support for other forms of synchronous communication, such as chat, or lighter weight capture capabilities, would have been useful for this type of group. Ju et al. have designed a capture system, called Workspace Navigator, for a similar domain (Ju, 2004). Student project groups used a dedicated physical workspace, which was instrumented with cameras for snapshots of the room, whiteboard, and physical objects and also recorded screenshots from the computers. Students used the system to aid in recording and reusing their design information. The capture occurred without any effort on the part of the students, allowing them to discover the usefulness of the record at a later time.

4.6.5 Ubicomp Research Group

The members of the Ubicomp research group have regularly used TeamSpace to record weekly group research meetings and occasional miscellaneous meetings since spring 2002, 125 meetings as of December 2004. A number of other smaller student meetings and talks have also been recorded. While capture was performed because of our motivation to use and research TeamSpace, the system has become regular, expected practice for group meetings. Thus, we are able to examine the use of the members of our own research group to provide insight into long-term practice and the motivations of meeting participants in reviewing meetings.

The Ubicomp Research group's meetings generally consist of one faculty member and 6 to 12 graduate students. The purpose of the meetings is for group discussions, feedback, and awareness on members' research activities. Meetings often begin with brief announcements, event planning or group coordination. One or two graduate students then lead the remainder of the discussion, giving practice talks, presenting ideas for discussion, and garnering feedback on paper drafts. The meetings are casual and free form, generally lasting one and a half hours.

Initially, we installed MeetingClient on a conference room computer hooked up to an electronic whiteboard. However, we rarely used the whiteboard, and MeetingClient was often simply run from a laptop. Additionally, in Fall 2003 we moved buildings and no longer had a SmartBoard in the conference room. Meetings were almost exclusively created and captured by the author on her laptop using an inexpensive microphone, although several group members have captured meetings as well. Thus, our use of

TeamSpace rarely required any interaction or effort from other group members, except for the author who functioned as a scribe. Capture and access research is a common theme in the group, so there never seemed to be any discomfort or objection with recording our discussions. Visitors were informed of the recording whenever they joined the meeting. Group members were shown how to use TeamSpace whenever they desired. Most learned how to use MeetingViewer when participating in the access behavior study presented previously, although several learned to use it on their own. Several members also learned to use MeetingClient separately.

Meetings generally contained 2 to 4 agenda items. The scribe also became the keeper of the agenda, keeping track of who was supposed to talk during the meetings and leading the meetings when the faculty member was not present. During meetings, the group created an occasional action item, 41 in total. The items were a convenient reminder mechanism since the item details are emailed to those responsible after the meeting. Any presentations given were loaded into TeamSpace. Since the speaker did not usually use MeetingClient to present, the scribe advanced the slides in correspondence. The scribe made a few annotations when appropriate and took spare notes for each meeting. She also checked attendance, although often forgot to note when people came or left in the middle of the meeting as her attention was on the meeting itself. We have felt that using MeetingClient has become an easy habit and does not normally detract from meeting participation unless a problem occurs.

Since most of the capture was performed by the author, we will focus the remainder of our discussion on how and when group members accessed recorded meetings. The results are not encouraging. Using the software logs, we found 25 uses of

MeetingViewer by 9 users other than the author. (There were also 7 cases of testing MeetingViewer or audio functionality that we are not including.) However, these are just the cases when a user was able to actually launch MeetingViewer. There are approximately another 25 attempts to review a meeting by 16 users that did not result in starting MeetingViewer. This occurred when the user had not downloaded the special class file that MeetingViewer required at the time. When the user inquired about this, the problem was fixed and resulted in a later review. However, many users appear to have just given up.

In responding to the question about why the meeting was being reviewed, 7 provided no answer. Five chose "Missed the meeting", five chose "Just browsing," and seven chose "Review a presentation." Three added their own options: "just checking on it", "relisten to the meeting", and "was late and wanted to listen to specific portions I missed."

The meeting access was only tried by two users prior to when the access behavior study began. Use increased immediately after an access behavior participation and was spread throughout the rest of 2003 and 2004. Of the 25 sessions, 2 lasted fewer than 30 seconds. In both of these cases, the meeting contained no artifacts and the user found empty information panels. The user quickly ended review of that meeting. The remaining 23 review sessions lasted anywhere from 48 seconds to 50 minutes. For some of the longer ones, the user appears to have been interrupted or doing other things, as there are many minutes of no activity. Thus, session length is not always informative. Several sessions were repeated reviews by the same user of the same meeting, likely due to problems that occurred when using MeetingViewer.

The most prominent thing that the MeetingViewer logs reveal is that playing the audio was problematic at best. While no one had difficulties in the access behavior study, almost everyone seems to have had issues on their own machines. In 20 sessions, users attempted to play audio. First, users often did not wait for audio to load, which could take a minute even without any other difficulties. Several users exited MeetingViewer without waiting long enough. Audio problems seem to be even more prominent in 2004, which could perhaps be a problem with the age of the TeamSpace software, and its incompatibility to components on users' machines. In these cases, the audio just never loaded, despite a few users trying for 10 minutes or more. Problems playing the audio seem to have plagued at least 13 of the review sessions, and audio eventually played in only one of those instances. Thus, users likely gained little benefit from using MeetingViewer in at least 12 review sessions. Users also did not report this problematic behavior, and instead gave up. Thus, the problems continued without awareness of the TeamSpace developer.

Two review sessions appear to be users verifying that capture occurred properly, as the same user that captured a meeting was immediately reviewing it. Again, 2 uses lasted less than 30 seconds, where clearly nothing was gained. 12 had audio problems where benefit was minimal at best. This leaves only 9 successful uses with possible benefit! Of those, 3 users did not play audio and only viewed the information panels. Useful information was contained in the notes or slides for these meetings. Thus, the interaction in these cases was very limited, but possibly useful.

We will highlight two review sessions that contained a good deal of interaction. Both involve the same user. The first, visualized in Figure 4.16, was to review a meeting he had missed. He navigates non-linearly using all of the artifacts. This demonstrates the benefits of all of these the indexes, which aided someone who was not in attendance to focus just on the topics that interested him. He spent considerable time listening to one portion of the discussion, then jumped around and briefly listened to other topics of interest, including one agenda topic and two points where action items were created. In the other meeting, Figure 4.17, he reviews a meeting where he gave a practice talk. Initially, he had trouble playing the audio. However, after 5 minutes of waiting, audio did successfully play, at which point he listened to the discussion that occurred at the end of his talk. Unlike the behavior seen in the access behavior study, in several of the successful review sessions, long segments of audio were played. Very little navigation

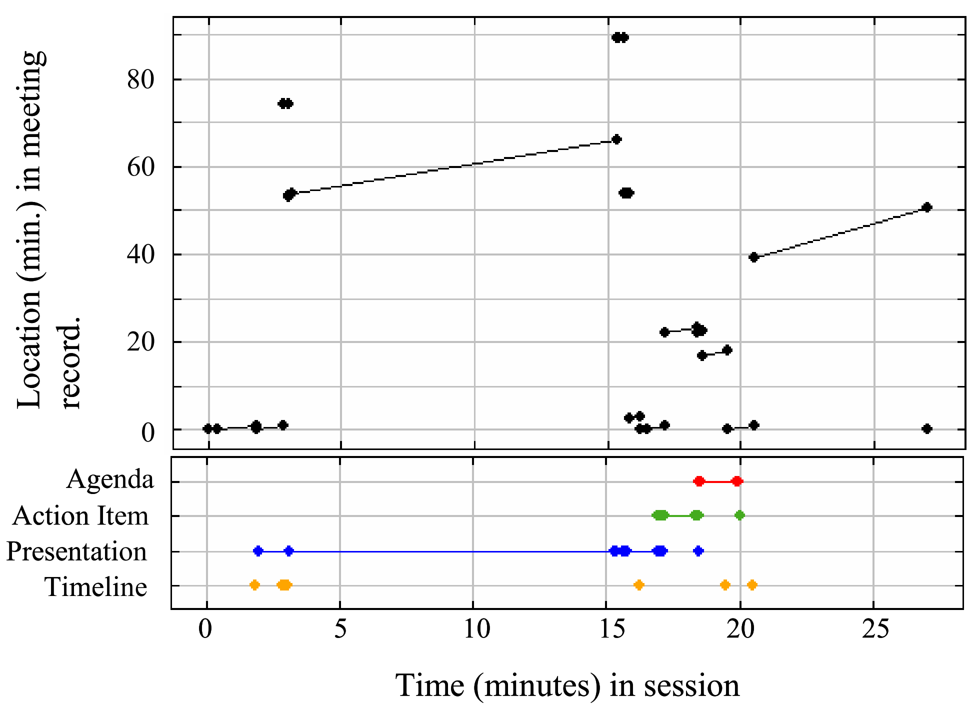

Figure 4.16. Review of a missed meeting.

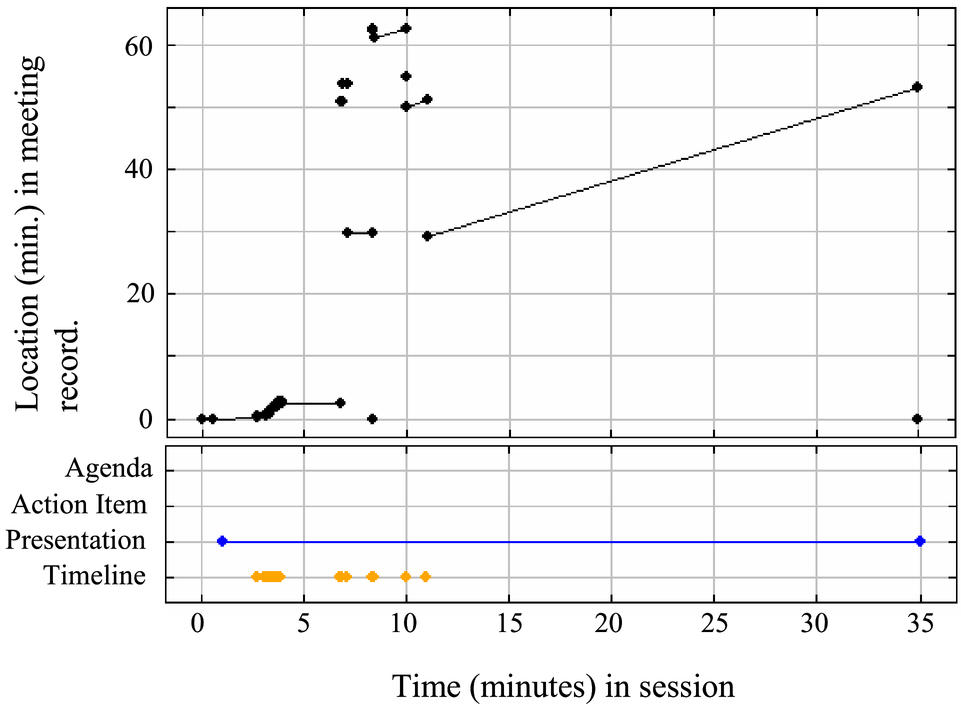

Figure 4.17. Review of a practice talk.

occurred once the start of the segment was found. Thus, while users may have been less patient for audio to load in their own uses, they were more interested in listening to long segments of audio. As we have so few instances to draw upon in this evaluation, we cannot conjecture how prominent skimming would be in other group uses. However, this result does imply that perhaps skimming may be less important than other behaviors.

All of the above use does not include the author, the only expert user of the system in the Ubicomp group. While analyzing one's own use is not particularly informative, of all users, she was most able to incorporate TeamSpace into her daily practice and came to depend on the system. TeamSpace was also easier for her to use, with fewer startup costs. So she was more likely to use TeamSpace when desired. The author marked her "real" accesses from all of her other tests, development, and research uses by answering the MeetingViewer questionnaire only in those real cases. There are likely several uses that went unmarked. As of December 2004, she used MeetingViewer 17 times throughout the years of deployment. In one instance she had missed the meeting, and in another she listened to a small portion of the audio where she left the room. However, her remaining uses contrast the usage by other members of the group. Besides the group meetings, she also recorded private and other important meetings occasionally. Thus, three accesses were to review the entirety of one of these meetings, a discussion going over the slides for an upcoming important talk. She listened to this discussion from start to finish, getting interrupted at various points and returning later to listen where she left off. For all but one of the remaining instances, she indicated that she was reviewing to "Find detailed information." She played audio in only one of these instances, where she replayed a small segment of the meeting multiple times. Thus, in all the rest, she viewed the notes or slides to look for the information she desired. Several times this information was annotations made on slides of her practice talks. In all the rest, the information was contained in notes of the meeting.

The reason we include these statistics is to note the contrast in use. First, for the author, the cost of starting TeamSpace was minimal. This lower cost likely contributed to her additional use. She also recorded more private meetings that she wished to remember, providing her with more access opportunities. Second, the author was usually the one responsible for taking notes and managing the agenda of the group meetings. As such, she was more aware what information was there, and that such information even existed. Thus, she was able to use MeetingViewer to look up small details that others did not. She actually did not make much more use of the audio than the rest of the group members.

These rare uses give us little to draw upon to compare the behaviors with those found in the access behavior study and discuss the usability of MeetingViewer. Scanning and jumping behaviors regularly occurred. And every type of index was utilized at some point. The artifacts were mainly useful for their indices, except for the one expert user. The artifacts may be more useful to other groups with higher information needs where more content is captured within the them. Lowering the cost of access will also enable these artifacts to be more useful. Private notes may also help access, as they serve as a reminder of the capture and are more personally meaningful to the users. In essence, the notes taken in the Ubicomp group meetings were the scribe's semi-private notes.

The review activities exhibited by the Ubicomp group did not involve searching for one particular detail in the audio. The group is co-located and in contact with each other frequently throughout the day. Thus, it is easier for us to ask each other than to consult the meeting record to find the desired information. Most of the access instead centers around listening to discussions of feedback that would be hard to recreate. Reviewers performed very little skimming, and were likely to listen to long segments of the meeting. For this type of meeting, methods to speed up this playback, such as audio compression, may be more important than features to improve skimming and replay. Additionally, this kind of important feedback was only useful to the one person it was given to, reducing needs for users to replay very often and increasing the costs of running MeetingViewer as more time passed between each use.

4.6.5.1 User Perceptions

Current group members were interviewed about their use and opinions of TeamSpace at the end of fall 2004. Despite the rare use of recorded meeting information and all of the audio problems encountered, group members were generally positive about

TeamSpace. As the use logs have already shown, most people admitted they had only made use of MeetingViewer one or two times. They felt that in general, there was little need to go back and review any part of the meeting. However, most thought listening to their own practice talks would be the most useful activity, and several did indeed do this. Several also commented on the benefits of the action items, which are emailed as reminders. Two group members mentioned that they were not as concerned about writing down comments or taking notes as they knew the meeting was being recorded. Thus, TeamSpace has led to a reduction in personal notes for some people. Others indicated that recording has not changed their practice at all. Many group members stated that they liked knowing that they could go back to any part of the meeting if they ever needed it. The fact that this was so rarely needed has not made anyone think that recording is not worth doing.

No one felt that recording had negatively affected the meetings. Most said that the microphone placed in the middle of the table provided enough indication of recording. While group members often joked that something just said was being recorded, no one felt that they had changed what they said or should need to. Everyone was aware that the audio could be muted if needed as this had been done in a recent meeting. But we suspect that several may have not been aware of this feature prior to that as it was only used a few times prior. Interestingly, one group member did not realize how many meetings had been recorded despite the occasional reminder in the meeting, and may have taken advantage of the content more if he had been aware. He was not involved in any of the early MeetingViewer studies, which may have led to this lack of awareness. Few expressed interest in recording other meetings. They felt that their current methods were sufficient. Although, three group members did express interest if the capture were more ubiquitous – something that was always with them and very easy to start, similar to the Personal Audio Loop application other members of the group have researched (Hayes, 2004).

The most encouraging aspect of the interviews is that every group member was interested in continuing to use TeamSpace and record the group meetings. The comments ranged from “Sure, why not?” to “yes, yes!” For all of these members, the capture is essentially free. The capture system is run by the scribe. Group members’ only tasks are to send her their slides whenever there is a presentation. Thus, even with such little need, the ability to have this safety blanket of recording is desired when the cost is low enough. Finally, most expressed that they did not feel that they had utilized TeamSpace as much as they could have. The cost of running MeetingViewer was usually too great to bother with. Even with our extraordinary measures to make this system easy to use and very robust, it was not enough. Additionally, even once that barrier has been crossed, group members are so rarely exposed to, and thus reminded of, the recordings and the information contained within them that they do not think to either capture or review except in higher need situations. More uses occurred following the access behavior review sessions, and for meetings of practice talks where the speaker was often reminded that they could review. Thus, the capture occurred in the background of the users’ context, and was rarely brought to the foreground where they could consider using the recorded information. The more frequent and quick uses displayed by the scribe, who frequently concentrated on capture and access, did not occur for the other group members. These uses show potential that has still not been fully realized.

4.6.6 Summary of Contributions

Our numerous attempts at deploying TeamSpace demonstrate the challenge of utilizing meeting capture in low need settings such as research and status meetings. We were correct in using a low technology, flexible capture implementation. The system was used in multiple meeting rooms and offices, and could be used in new environments with little extra effort. A smarter meeting room may have improved the capture quality, but the quality was not the problem in our experiences. Instead, users considered the cost of learning and using the capture and access software too high given the relatively small benefits they provide. In fact, while everyone was much more positive about the concept of meeting capture than we expected, they used the system much less than we expected due to this cost. With essentially no cost involved, all of the members of the Ubicomp group would continue to capture their meetings and we have continued doing so. Other groups could get similar use with one motivated person incurring the capture costs for the whole group. Systems that automatically capture information without any user intervention also lower this cost, but at the expense of possibly limiting the indices that can be captured, and bringing capture to the attention of the users. Additionally, improved integration with existing tools and activities could continue to lower the costs of capture.

A number of improvements can also decrease the costs of access. First, users need to remember that the information exists and how to access it. While the emailed action items provided such a reminder, other types of frequent reminders would also help. Additionally, as the research scientist Brian noted, having to go through the web interface was often cumbersome. TeamSpace needs to provide faster ways to directly launch meeting review. Finally, systems that would contain personal information may also provide the buy-in necessary to gain added use. Robustness is also still a big issue for TeamSpace despite great development efforts, and complicated research prototypes may always have such difficulties. However, even with all of these improvements, reducing the cost of access to the near-zero levels that these low need situations may require will be challenging.

Some of the costs involved in using TeamSpace resulted from use by groups that we did not explicitly design for. The distributed TeamSpace development team needed some support to facilitate meetings. Thus, we were willing to expend some effort in using a tool, with the capture being a nice feature of that tool. Additionally, the group workspace functionalities of TeamSpace were meant to encourage usage of the system for many things, such as group calendaring, document sharing, and event awareness. The groups who used capture did not have a need for such features for the most part. Instead, needing to go to the group workspace to perform any capture and access function was a burden. And the meeting record was not at all integrated with users' work processes since those were all external to TeamSpace. To more fully explore capture and access in this context, we recommend integrating capture capabilities with an existing, robust collaborative workspace system and recruiting users who already use that system. This will eliminate the need to solve all the issues and adoption of a collaborative system (Grudin, 1994) while also trying to investigate capture and access, and would potentially lower the costs of use sufficiently. Our experience also demonstrates the importance of designing capture and access around the needs and work context of the users. Our attempts to keep the costs of use low did not work as well when those needs were somewhat different than we anticipated.

We have perhaps painted a rather dismal picture of the use of TeamSpace in these studies. However, there are positive things to note. The first is that, even at Boeing, every person who was interviewed was rather positive about the prospect of recording meetings. Few expressed privacy concerns at this point. Those concerns may only arise when such systems are widely deployed enough to be abused occasionally. Despite a few problems, TeamSpace was able to be used by a variety of people over a period of years. This allowed us to learn from even infrequent use, something other meeting capture researchers have not been able to do. And, while review was rare, it did occur and provide at least some benefit to those users. Improvements to TeamSpace would increase the review that occurs in the situations we have examined, even if that use remains somewhat infrequent. These improvements mainly involve tuning the system to the work context of the users to aid them in accessing the recording, and require researchers to focus more on access motivations, behaviors, and interfaces, and less on the capture technology. With a more evolvable approach to implementing and deploying such services, other research systems could improve upon the lessons of TeamSpace and continue to examine the benefits of capture and the situations where such technology is wanted and needed.

4.7 Conclusion

With TeamSpace, we succeeded in implementing a scalable, robust, and usable meeting capture and access system that could be deployed in a number of environments. We have shown how indices can be created using general meeting artifacts, and that these indices were indeed important in accessing the meetings. We have conducted a detailed evaluation of the access interface, gaining an understanding of a number of low-level behavioral patterns involved in browsing and searching captured meeting information. Our lengthy deployment allowed us to learn something about the motivations for use, and the challenges in using such a system. Despite the low level of real use, the deployment and the feedback from all of our users still show potential benefits for meeting capture and access that we have not yet met, and show that users are open to those possibilities. We further discuss the issues that were raised and the lessons we learned in this, and all of our prototypes, in Chapter 6.

CHAPTER V

KNOWLEDGE ACQUISITION

In our TeamSpace deployments, we found that users have few needs to remember much of the content of everyday meetings, resulting in few benefits and little use of capture and access systems. Yet there are higher need discussions, such as the SAAM discussions, whose details may be more important to document. Thus we wish to look more closely at meetings in contexts where the content of the discussion is clearly important to remember and document, and capture can have a potentially bigger impact on the use of that content. We also would like to focus on known outcomes that can be evaluated. To that end, we have examined capture and access of knowledge acquisition (KA) sessions, discussions involving systems designers and domain experts for the purposes of understanding the domain, the problem, and user needs. The knowledge learned from these sessions is then refined into domain models and requirements specifications that feed design and development.

For complex domains, such as military systems, the sheer size of the knowledge base is impossible to be managed and controlled by a single individual. Additional problems stem from knowledge being gathered from many different sources by different individuals, often in contradiction with each other. This knowledge can be used in non-obvious ways in design and is rarely traced through the whole lifecycle of design, usage, test and evaluation, and re-use or modification. This can result in misuse of knowledge and duplication of effort gathering knowledge over the lifecycle of the project. Many of these problems may be aided by improved records of, and access to, the original KA discussions throughout development.

Smart Information Flow Technologies (SIFT) proposed the concept of annotating knowledge acquisition sessions with various concepts to aid in the capture of knowledge and the transition of that knowledge to further software development phases (Richter, 2004). We have worked with SIFT on prototyping and evaluating TAGGER, a tool for capturing and indexing these sessions. In addition, we have implemented an access prototype and used it to evaluate the potential benefits this capture method can provide.

5.1 Knowledge Acquisition Process

The lifecycle of knowledge in a large-scale system begins as knowledge is acquired and captured (recorded) through knowledge acquisition (KA) sessions. Knowledge is then passed along to software development efforts that, among other implementation activities, must translate "raw" knowledge acquired in natural language transcripts or KA session notes into whatever representation is desired and ultimately used in design and implementation. Several rounds of testing for validation and then use for verification may follow, with subsequent knowledge tuning phases. From any of these steps, an evaluation of the existing knowledge can be performed and new knowledge requirements can be identified, which restarts the knowledge cycle.

Knowledge acquisition (KA) sessions consist of a knowledge acquisition team that interviews one or more subject matter experts (SMEs) about their domain. The goal of the knowledge acquisition team is thus to record all of the important information revealed in these interviews. To gain a better understanding of KA problems, SIFT studied existing knowledge acquisition literature and interviewed KA members of the U.S. Army's Rotorcraft Pilots Associate (RPA), an extremely large and knowledge intensive project (Miller, 1999). They found that using advanced recording and document management tools, the RPA team captured huge amounts of verbatim data—video, audio, transcription and some processed documents. Various attempts to structure this data after the fact were seen as not useful both because they were time consuming and because they did not provide the organization that software engineers needed or wanted. Indeed, the single biggest problem in knowledge acquisition for RPA was the useful transfer of knowledge from KA to software development. The rich body of information acquired in KA must be reduced to a compressed set of requirements for further software development. This is inherently a lossy process; much of the knowledge about why requirements are as they are, and how they might be changed, is removed. This problem manifested itself in a variety of ways on the program, including:

- KA was seen as little or no use to development.
- Large amounts of data were captured but never looked at again.
- Software development had problems getting answers to their questions.
- Duplication of effort by KA teams.
- An inability to trace and understand the rationale of requirements.

Worse, as the RPA program attempted to transition its technology to new applications, subsequent developers found it difficult to understand why specific requirements and design decisions were in place.

5.2 Tagging

What designers and developers need is a means of tracing the lifecycle of every bit of knowledge in the system, and of every design decision that involves that knowledge, in such a manner that does not impose undue additional workload on the engineers who must build the system. To be effective, this mechanism must be integrated with the other software development activities underway. This means taking knowledge initially captured in response to requirements, and facilitating its transition to representation and implementation. This may, in turn, require greater communication and sharing of representations between KA and software development.

One critical link is a capability to tie the knowledge captured in KA, the mounds of video, transcripts, and documents, into the structures used in software development. This would create traceability not only from implemented software to an initial requirement, but also from the requirement to the set of (potentially distributed and contradictory) statements that were used to produce that requirement. Such traceability would enable the requirements engineer, designer and developer to resort to source material—that is, to the initial knowledge acquisition sessions—when needed. It would also potentially enable these software developers to communicate more precise knowledge needs to the KA team, and help the KA team know when contradictory and/or missing knowledge must be resolved. In other words, by capturing and providing meaningful ways to access knowledge acquisition sessions, we can aid in the traceability of knowledge throughout the software life cycle.

The first and most critical step in providing access to any captured discussion is to ensure those discussions are structured into meaningful units and indexed with useful identifiers. Our approach is to take advantage of the artifacts and activities of the discussion. Yet, standard artifacts are not commonly used in knowledge acquisition discussions. Instead, we focused the structures that fit the use of the information – those that are common to software development. While few formal structures exist at this stage, we chose several simplified structures that can turn into more formalized representations throughout development. As few of these structures can be automatically recorded or interpreted from the conversation, we chose to examine user-driven indexing, a process we call "tagging."

"Tagging" is the concept of assigning one or more identifiers to a word or phrase in the dialogue, thus creating a semantically meaningful index. To create this structure effectively and efficiently, we argue that the initial identification and linking of these indices should be done in real time – during the KA session. This ensures that those already participating in a KA session are helping to structure the captured knowledge, and requires no additional time for the KA team. Even if done after the fact, efficiency is still important to encourage use. Thus, the complexity of these indices must be limited, yet still meaningful.

This method of indexing is different from a general theme in meeting capture not to drastically change the activity being recorded. This is the approach we took in both TeamSpace and SAAMPad in matching the capture to the existing activities. However, in this case, there are few indices we can automatically capture and derive. Additionally, KA team members already attend these sessions in order to take notes and capture the knowledge in the discussion. Thus, while their method of doing so may be different, we are not changing the purpose or goals of the KA team members. The effort to produce these indices may be higher than in many systems, but we believe the potential benefits of such semantically meaningful indices will be worth that effort. Additionally, this provides the opportunity to examine a very rich and meaningful set of indices that we cannot do in other domains.

We derived our tags based on an examination of KA problems, existing structures and uses. Table 5.1 summarizes the set of tags we propose. Our tags are based upon *domain specific*, *domain independent* and *conversational* structures. *Domain specific* tags denote the specialized terminology of the domain and the unique aspects of the environment. *Domain independent* tags are based on the structures of object oriented programming and design. Their aim is to help form the model of the project domain. The goal of the *conversational tags* is to help capture the structure or flow of the conversation and record the arguments and outcomes of the discussion. These tags are derived from the rationale capture annotations described by Conklin and Begeman (Conklin, 1988). Each of these categories has more specific tag types that are used to annotate the discussion. We believe that this particular set of indices is a useful structure for KA sessions, and have used them in building and evaluating our prototypes. However, we do expect that such indices will evolve or be customized for different groups or situations, and that similar user-driven structuring would work in other domains. In Table 5.1, we summarize the set of tags that we are proposing to structure a KA dialogue. In the following sections, we define each tag and present them in an example discussion. While the example in Figures 5.1, 5.2, and 5.3 does not contain every tag type, the dialogue is realistic of a KA session and provides an example of a type of problem that can occur in knowledge acquisition.

Table 5.1. Proposed tag set.

1) Domain Dependent Tag Types
a) Thing
b) Actions (Tasks)
c) Conditions
2) Domain Independent Tag Types
a) Classes
b) Attributes (associated with Classes, characteristics of the class members)

c) Operations (associated with classes, actions the class members can take)
d) Associations (exist between classes; default is generalization.)
e) Multiplicity (tied to associations between classes)
3) Conversational Tag Types
a) Issues (= Questions, can be associated with other issues or with positions or arguments)
i) Can have status = Open (to be discussed further), Closed (resolution has been decided), or Tabled (closed without resolution)
b) Resolutions (= closure for an issue. Becomes a requirement in the case of total acceptance or rejection.)
i) Can have status = consensus (the group generally agrees with this rationale) or partial (some agree and some disagree) or rejected (all disagree).

5.2.1 Domain dependent tags

Domain dependent, or domain specific, tags denote the specialized terminology of the domain and the unique aspects of the environment. These tags will indicate something the developer likely needs to understand, and may become an important reference term for finding relevant passages in the future. Domain specific tags can be of three types: *things*, *actions*, and *conditions*. *Things* are used to denote concrete nouns in the domain, terminology that is new or used in new ways. *Actions* are unique tasks or activities performed in the domain. *Conditions* are areas of dependency, where different decisions or facts would hold under different circumstances. In the example in Figure 5.1, military slang like “friendlies” and “enemy units” are tagged as a *thing*. Other terms, such as “mobility projection areas” or “movement areas” are also important objects of this discussion. Additionally, SME3 says that under certain conditions, namely wearing Night Vision Goggles, the use of red may be problematic. Thus, Night Vision Goggles is marked as a *condition*. It is important to note that this example is meant to convey how tags might be applied, not the appearance of the actual user interface.

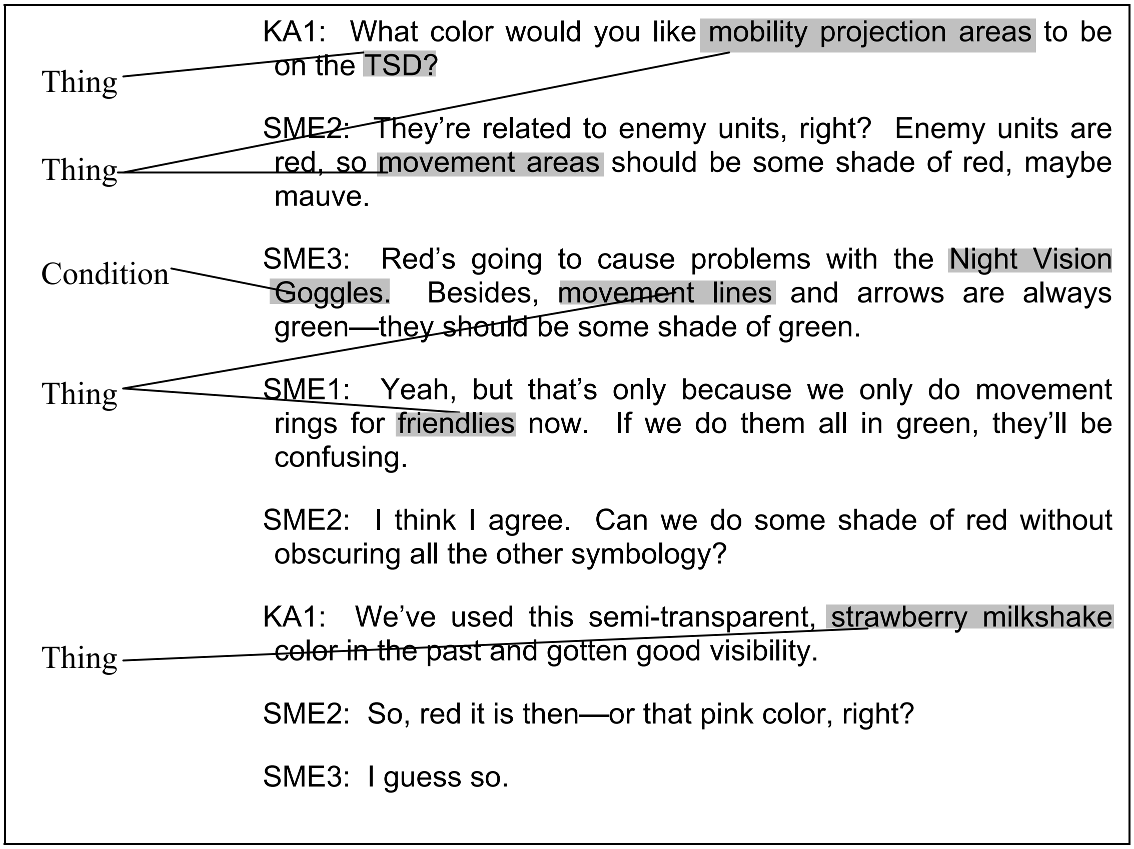

Figure 5.1. Domain dependent tag example.

5.2.2 Domain independent tags

The domain independent tags are based on the structures of object oriented programming and design. The aim, among others, is to help form the model of the project domain. Thus, the tags are *class*, *attribute*, and *operation*. Two other tags, *multiplicity* and *generalization*, describe the relationships between classes. *Classes* are the objects of the domain, objects that one would expect to instantiate within a system. *Attributes* are characteristics of a class, including specified ranges or default values of such descriptions. *Operations* are actions that a class may take, things that can be done with or to the entities represented by a class. As attributes and operations are characteristics of a class, there will also be explicit relationships marked between tagged attributes, operations, and their associated classes. Additionally, classes can be related to each other in various ways. One is *generalization*, showing inheritance or the “is-a” relationship between two classes. Additionally, *multiplicity* is tied to associations between classes and indicates how many objects may participate in the association.

There is likely to be a correlation between these tags and the domain specific tags as domain dependent entities are instantiated in a domain model. For example, *objects* and *attributes* may correspond to *things* while *operations* are likely to be *actions*. Additionally items tagged as *conditions* may correspond to attributes of relationships between objects.

In Figure 5.2, “mobility projection areas” and “TSD” have been tagged as classes while “color” is an attribute of mobility projection areas. The colors “red” and “green” are also attributes, specific colors that are being discussed. The relationships between classes, and between the attributes and classes, is shown as a line between tags.

We don’t expect the KA team to build class diagrams on the fly during a KA session, but instead note phrases that will likely relate to these future models. The simple tagging of concepts that occur during discussion with some or all of the syntactic structures that software development provides potentially improves traceability of these concepts back to the original experts.

5.2.3 Conversational Tags

The goal of the conversational tags is to help capture the structure or flow of the conversation and record the arguments and outcomes of the discussion. These tags are derived from the rationale capture annotation IBIS (Conklin, 1988).

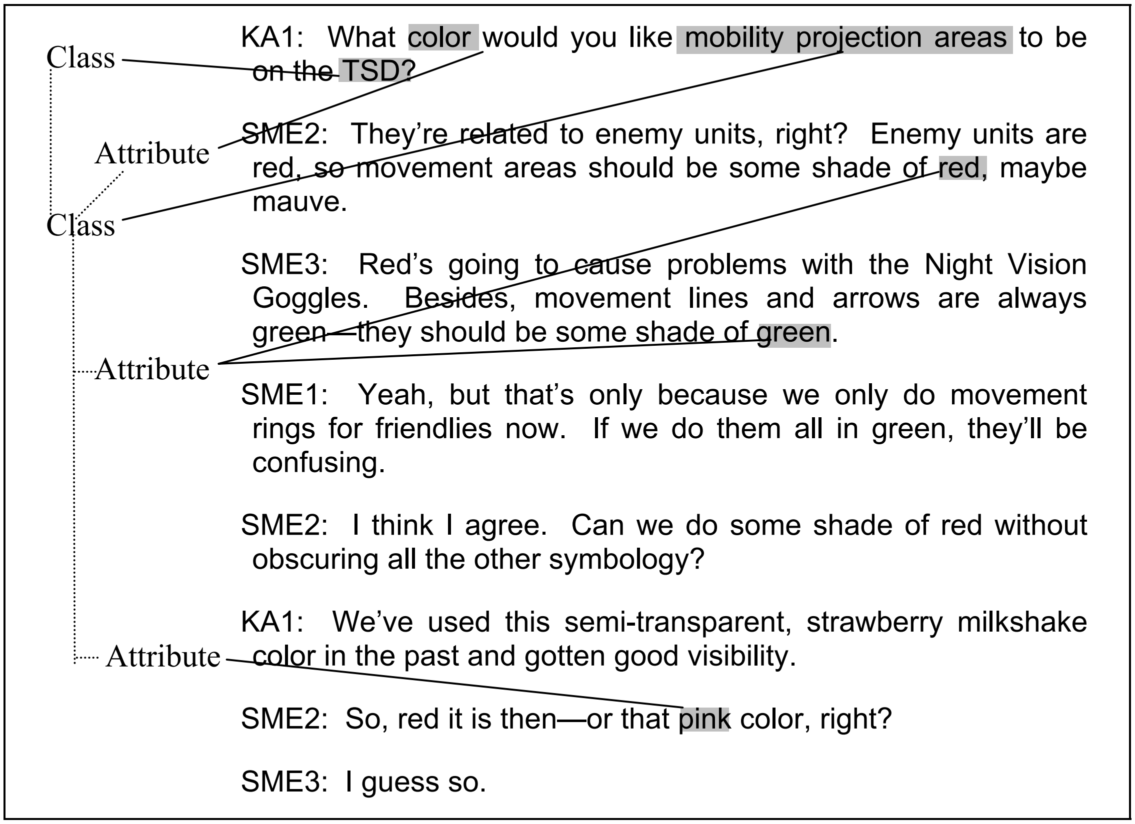

Figure 5.2. Domain independent tags example.

An issue is, basically, a topic of investigation or discussion. In knowledge acquisition sessions, issues frequently correspond to questions. Each issue tag has a status associated with it. The status can be:

- Open—currently being discussed.
- Closed—a resolution has been decided and the issue is not expected to produce further discussion.
- Tabled—discussion of the issue has been abandoned without resolution.

Additionally, issues can be related to each other, as one may follow on another, or be a sub-issue of another issue.

Resolution tags are used to mark the end or closing of discussion on an issue. Thus, they should be linked to a prior issue. Resolution tags also have a status. The status is used to reflect the tagger's opinion of how universal the agreement was on the resolution. They are:

- Consensus—the group generally agrees with this outcome
- Partial—there is majority agreement, but some hesitancy, reluctance or partial disagreement.
- Rejected—there is no general agreement to this outcome. A significant number of participants hold different opinions.
- Null—for use when the tagger is not certain how much agreement there was on the outcome, or when the resolution was offered for a topic that was abandoned or tabled.

In the example in Figure 5.3, the conversation revolves around the issue of what color to make the mobility project areas, with a sub-issue of whether some shade of red would work. A shade of red is suggested, but is only partially agreed upon.

5.3 TAGGER

"Tagging" is the concept of assigning one or more of the tag types to a dialogue of a knowledge acquisition session. To tag effectively and efficiently, we argue that the initial identification and linking of important information should be done in real time – during the KA session. In particular, this ensures that those already participating in a KA session are helping to structure the captured knowledge with no additional time involved. Some amount of structuring and organizing of recorded KA audio or video is certainly performed as a part of KA activities today—but almost always after the fact when some usually junior member of the KA team is required to wade through hours of captured information to derive those portions deemed relevant. This process is inefficient and

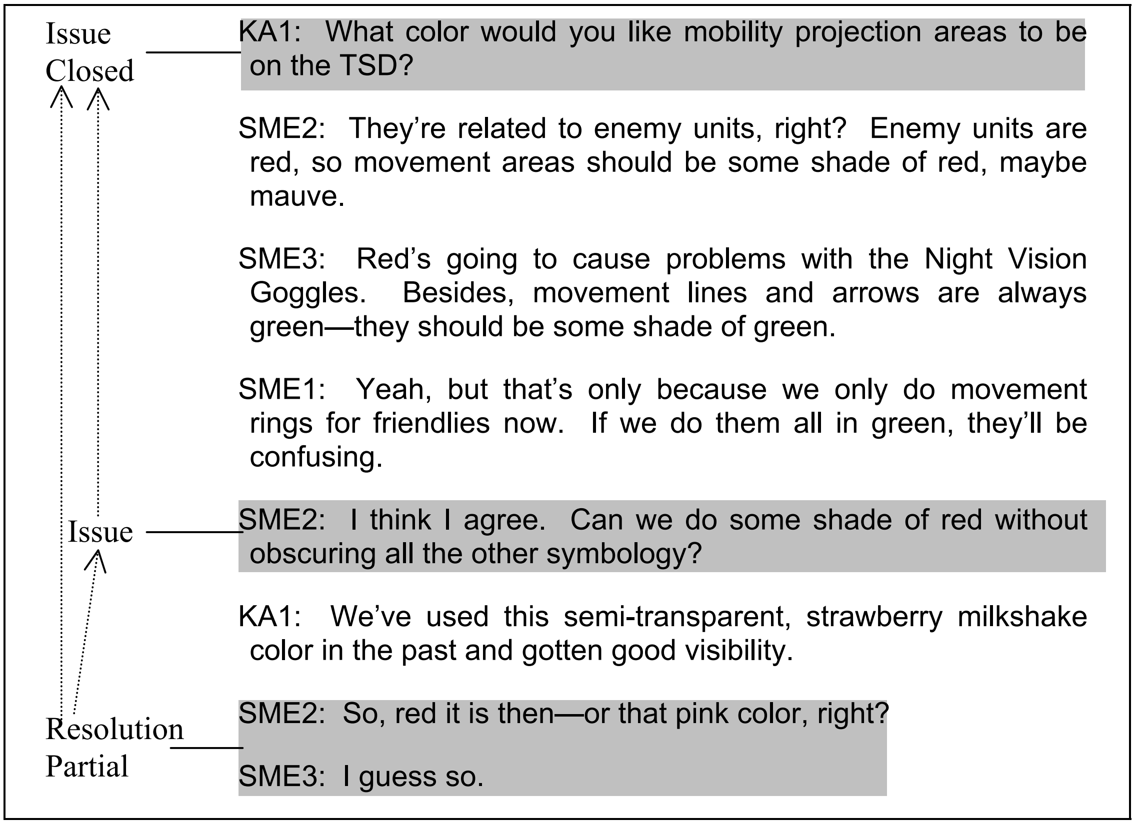

Figure 5.3. Conversational tags example.

lossy, and often does not occur. Instead, we believe we can ensure tagging richness and comprehensiveness by enabling multiple knowledgeable participants to do that tagging from multiple viewpoints, simultaneously, during the KA session.

SIFT has developed the prototype TAGGER (Team-Aware Acquisition Guide for Goals, Entities, and Relationships) to capture and record knowledge acquisition sessions, and enable tagging of that recording in real time. The system automatically records audio and video, transcribes the audio, delivers the transcript to individuals with laptop or tablet computers to perform tagging, and synchronizes all of the streams of tagging and input. We expect that different users will be responsible for tagging different tag categories, thus reducing the amount of tagging required of any individual, yet allowing multiple viewpoints to be tagged. Thus, the indices we are exploring with TAGGER are all user-defined and asserted. While this resembles video annotation, discussed in Section 2.2.1.1.1, we are examining both real-time tagging of conversation, as well as the use of certain categories of tags that are particularly useful in this domain.

In designing and building TAGGER, we examined the feasibility of this real-time user-driven indexing. With several small experiments, we evaluated whether users were able to recognize and annotate the various concepts, their understanding of the different tags, and their ability to perform this task realistically (Richter, 2004). In the following sections, we describe the studies and their results.

5.3.1 Evaluation: Understanding Tags

The tags outlined above were created and refined by examining KA problems and software development structures. However, these tags need to be put into use by taggers other than ourselves. In designing TAGGER, our first goal is to understand users' behavior in tagging real KA sessions and their understanding of the various tags. These results will provide data to serve as a baseline for understanding the performance of future TAGGER prototypes and may signal the need to modify the proposed tag set as well.

We performed a small study using six subjects, computer science graduate students at Georgia Tech. These students have backgrounds similar to that of a junior member of a KA team – a graduate in computer science with some experience in development. Each subject was given a training document explaining knowledge acquisition sessions, the tagging concept, and details and examples of each tag. They were then given a paper transcript of a knowledge acquisition session to tag by marking up the transcript.

The recorded KA session was an hour-long interview between one SIFT member, who has significant KA experience, and two airline pilots. The interview focused on take off procedures and checklists with the purpose of designing an electronic checklist system for the cockpit. While the session was not part of a real project, the interview itself is realistic, involving real pilots giving real answers to questions about their domain and a potential future system to be used in it.

From that KA session, we cut out several multi-minute dialogues on different topics to use for this study. Subjects were asked to read the transcript of these segments and tag it for one category of tags at a time. They then re-read the same transcript, and tagged it with another category. The order of categories was varied for each subject. Subjects tagged by circling or underlying tags, and using an abbreviation to mark which tag. To relate, or link, tags they drew lines between them. Subjects could take as much time as they wished to tag the transcript, as we wanted to first investigate tagging without the constraints of time involved.

5.3.1.1 Results

Subjects reported having no difficulty understanding the tagging concept, or any of the individual tags. Multiplicity was the least used tag, occurring only once in one subject's tagging. This may be due to a lack of applicable instances in the transcript, but it also suggests that removing multiplicity from our stated tag set may be warranted. Conversation tags were seen as most difficult to assign because their use spans greater distances in the transcript in order to track and link issues and resolutions. Additionally, subjects reported that with conversation tags it was more difficult to determine what was important to tag, and what besides a clear question was an issue. There was much variation in what was tagged, yet there were no clear mistakes or errors made in tagging. In other words, all tags asserted can be interpreted as reasonable.

On average, subjects asserted 7.5 domain-specific tags per minute of conversation, 10 domain independent tags per minute, and 2.3 conversation tags for a total of 19.8 tags per minute of conversation (over three passes through the same transcript). We considered relationships, or links, between two tags to be another tag itself. Tagging performance, however, varied widely, from 12 total tags per minute to 24. Some subjects seemed to limit tagging to items they deemed truly important for the topic of electronic checklists, others seemed to tag much more thoroughly.

The order of tagging did affect results. Those subjects that tagged domain specific tags first, tagged fewer domain independent tags, while those that tagged domain independent tags first, tagged fewer domain specific tags. Thus, there was overlap between things and classes, and between actions and operations, and subjects resisted tagging the same thing multiple times. Improved definitions, examples, and training of each tag may reduce the amount of overlap, and thus the potential for duplication of effort when multiple people are tagging the same conversation.

Overall, this study shows that tagging is an understandable process. Yet the results do vary — what is important and how to tag it is open to interpretation. This is not negative, as long as the knowledge acquisition team can have confidence that the session is being tagged reasonably and consistently. However, in the context of a real project, with a better understanding of the goals of the knowledge acquisition session, this variation may be reduced. Additionally, experience will also likely reduce variation as users learn how they utilize different tags.

5.3.2 Evaluation: Understanding tags in real time

Our next step was to examine tagging in real time. In the previous study, subjects had as much time as they needed to tag the dialogue, which resulted in many tags being created, but also in a significant amount of time being taken to tag the entire conversation. Participants usually took an hour to tag 11 minutes of conversation 3 times. In this study, we restricted users to tagging in real time, as the video of the session was playing. We expected that the amount of tagging would be reduced, and closer to what we can expect with a TAGGER prototype.

In this study, six different subjects were chosen from the same graduate student population, given the same training, and used the same tag set as the previous study. We used the same KA dialogues from the previous study along with several additional minutes. Subjects were asked to tag the transcript while they followed along with the video of the discussion as it played, with tagging ending a few seconds after the video did. Subjects performed tagging with one category at a time, and tagged three separate segments of dialogue, each approximately 5 minutes long, using a different category for each segment.

5.3.2.1 Results

As in the previous study, subjects again reported having no difficulty understanding the tags. Multiplicity was again seen as the least useful tag. As one subject stated, "There is very rarely a clear trigger in the conversation that prompts me to use multiplicity. I just didn't think much about it." Domain specific tags were seen as the most difficult to assign in real time. Relating classes, attributes, and objects required users to look back in the transcript, find the tag, link it, then catch back up with the conversation. Again, there were no tags that were clearly mistakes or errors.

Table 5.2 summarizes the tagging frequency results from both experiments, with column 1 showing the results from the previous study, and column 3 showing the real time results. Despite our expectations that tagging frequency would be noticeably less, subjects performed about the same as the average in the previous study. On average, subjects tagged 9.4 domain specific tags per minute of conversation, 9.8 domain independent tags per minute, and 3 conversation tags per minute. These numbers, in fact, appear as good or better! However, the segments chosen in this experiment were particularly suited to the type of tag category being used. We can directly compare the same dialogue for domain independent and conversation tags, although most of the dialogue used for the domain specific tags was not used in the previous study. These results are shown in column 2 of Table 5.2. In the non-real time study, the rate for the comparable dialogue is 12 tags per minute for domain independent tags. This is a better rate than in real time, although not statistically significant. Examining the transcripts shows that real time subjects tagged the same items multiple times, whereas non real-time subjects tended not to do this. Thus, there were fewer unique tags in the real-time case. However, this aided real-time subjects in that they were able to relate more things that were close together, possibly allowing the relation to occur at all as they did not have to look far to find and draw a line to the related tag. Duplicating tags is not necessarily negative as it provides additional indices into conversations about the same important entity or topic.

Table 5.2. Tagging frequency for all conditions. Values in bold font are significantly different from results of previous experiments.

	Overall Non-Real Time Rate	Comparable Non-Real Time Rate	Real Time Rate	Error + Delay Rate
Domain-specific tags	7.5	-	9.4	**5.1**
Domain-independent tags	10	12	9.8	**5.8**
Conversation tags	2.3	1.8	3	2.5

For conversation tags, the non-real time rate for the comparable dialogue is 1.8 tags per minute of conversation. This is actually noticeably less than the real-time subjects, although still not statistically significant (p=0.65). The real-time subjects tended to tag larger chunks, usually the entire statement of one person. Perhaps the time constraint, and this simplification, helped subjects look at the conversation at a higher level, and thus assert more tags.

The subjects in this study also had less variation both in their tagging frequency, and what they tagged, than the previous study. This is also not surprising, as there is an obvious upper bound on how much tagging can be done. Additionally, the time constraint likely led subjects to focus on what they felt were the most important tags, thus reducing the differences.

This study showed that tagging can feasibly be done in real time, with results that are similar to tagging without time constraints. The implication is that knowledge acquisition members already attending the KA session can produce an indexed KA recording that is similar to what could be produced without time constraints. However, subjects in this experiment felt that tagging was challenging, requiring a lot of thought. Thus, knowledge acquisition team members may not be able to keep up with such a high rate of tagging for a long period of time. Again, experience will likely make this task easier. We can also investigate the interactions that can minimize the workload of asserting tags.

5.3.3 Evaluation: Effects of Transcript Technology

In the initial studies, subjects tagged human-transcribed, error-free records of the KA session which had been prepared in advance of the trial. Any foreseeable speech-to-text transcription software will never be error-free and will, inevitably, introduce some time lag between an utterance and its appearance in the transcript. Thus, in our next study we looked at the effects of the errors and delays that automated transcription might introduce.

Again, we used 6 computer science graduate students as subjects, and they tagged the same dialogue in the same manner as the real-time condition previously. However, in this case, the transcripts were created using existing speech-to-text software, and were delivered on a tablet computer, in chunks of 2 to 5 seconds, with a delay of 1 second. Thus, subjects would see text anywhere from 1 to 7 seconds after it had been said. Due to high background noise in our original video recording, we produced the transcripts by reading back the exact transcript—including all of the original "ums," repeated words, etc.—into a version of Dragon's Naturally Speaking 7.0 that was not trained for the speaker. Some of the dialogue was read at a slower and clearer pace than the original video. Thus, our study represents something approximating a best-case scenario of what might be produced in a multi-speaker, individually miked, real time conversation in the not-too-distant future. The transcripts were approximately 75% correct, but with no punctuation. However, many of the errors were not in the distinct, domain-specific words, but in transitions or the beginnings of sentences, and thus, in words that were less likely to be tagged. Subjects circled words using the tablet pen and labeled them using abbreviations as in the previous studies on paper.

5.3.3.1 Results

The tagging frequency results of this study are shown in Table 5.2, the fourth column. Subjects in this condition averaged 5.1 domain specific and 5.8 domain independent tags per minute of conversation. Both of these are significantly different from the rates in the real time condition ($p=0.008$ for both). Subjects also averaged 2.5 conversation tags per minute, a rate not different from the real time study. In other words, performance was impacted for both domain independent and domain specific tags, but not conversational tags.

As reported by the subjects, the delay in seeing the transcript did not cause many problems, as subjects had difficulty keeping up as it was. However, the errors in the transcript did have a major impact. As one subject reported, the errors were “distracting,” increasing the amount of time to find and tag phrases, and did cause subjects to tag non-useful phrases by mistake in several instances. However, given that many of the words likely to be tagged were recognized correctly, we do not necessarily need a better recognition rate. Instead, punctuation, or some other form of phrase segmentation, may aid in matching what was heard with the text more easily, and thus, help users find the desired phrases to tag more quickly. So while performance was impacted by the errors in the transcripts, tagging was still able to occur at what we feel is a reasonable rate. We also need to investigate opportunities for some types of automated tagging interactions to minimize the workload of asserting tags.

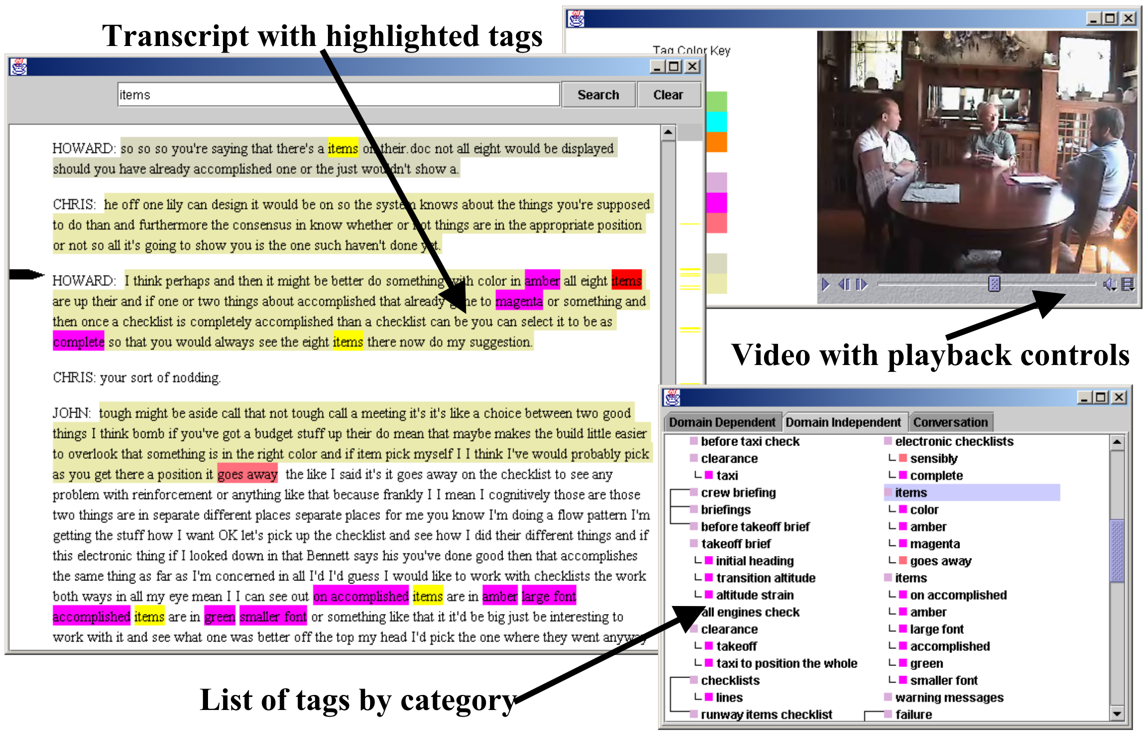

Figure 5.4. Screenshot of TagViewer. The lines in the tag list show class attributes and operations, and relationships of classes to each other. The tag "items" has been used to index into the transcript.

5.4 TagViewer

The purpose of capturing and indexing these sessions is to provide the KA team members, and later software developers, the means to find and review portions of the original KA conversation as needed. To demonstrate and begin examining this use, we created an interface prototype called TagViewer that allows users to view all of the tags of a knowledge acquisition session and use them to index into the tagged transcript and video. Our goal in designing TagViewer was to provide high visibility of the tags and quick access to the transcript and video with very simple interactions.

A screenshot of TagViewer is shown in Figure 5.4. The interface is composed of three windows. One contains the video and standard playback controls, such as play and stop, along with a key for the colors used to highlight the various tags. The second window is the tagged transcript of the interview. The tags are highlighted in various colors throughout. Clicking anywhere in the transcript text jumps the video to that point in the session. Thus, the transcript serves as a fine-grained index into the video of the KA session. A marker in the left margin of the transcript indicates the video playback location. Additionally, the transcript is searchable. Search words are highlighted in red, with the overview bar at the far right showing the location of the results in the entire transcript.

Finally, the third window in Figure 5.4 is the tag summary window. For each category, the tags are listed in the order of appearance in the transcript. For the domain-independent and domain-dependent tags, clicking on the tag scrolls to and highlights that tag in red in the transcript window. Additionally, a search is done on that tag, and all other instances of that word or phrase are highlighted in yellow, thus giving an indication of where else in the KA session that concept was discussed. Clicking on a conversational tag, generally several sentences of text, merely scrolls the transcript to that point. Thus, the tag summary window serves as an index into the textual transcript, while the transcript serves as an index into the video itself.

We informally tested the usability of the interface while piloting the study described below. The prototype was easy to use and sufficiently functional for our evaluation. There are many other features we can imagine adding to TagViewer to make it easier to use and more scalable, such as the ability to view multiple interviews, turn on and off tag highlighting, improved searching capabilities, and ordering and searching of tags themselves.

5.5 Evaluation: Benefits of Use

The previous set of experiments studied the feasibility of manual indexing, and we now turn our attention to the use of that indexing. We believe that improving the access to Knowledge Acquisition records benefits any stage of software development where that knowledge is used, from continued KA efforts, to development, and into re-use or modification. To begin examining these potential benefits we performed an experiment where participants needed to use the knowledge from an hour-long knowledge acquisition interview to create a requirements document. We compared a typical situation today, the use of notes and standard video, to the use of captured and indexed sessions to determine the value and use of capture above and beyond just video recording. We hypothesized that those using tagged transcripts would create better quality requirements documents than video-alone because they would have better and more efficient access to the original knowledge. This is the first such experiment examining the potential benefits gains of meeting capture that we are aware of. This study demonstrates how software engineers could utilize the captured KA sessions in immediate tasks, using tags to clarify or search for additional information in the KA session, and to browse for forgotten or additional issues. While the use of TagViewer in this experiment did not yield strong quantitative benefits, there were very clear differences in the use of the video record. Our results provide insight into the potential of such capture tools to improve access to important discussions.

5.5.1 Study Description

The study used 12 computer science graduate students at Georgia Tech. These students have similar backgrounds to a typically junior knowledge acquisition team member – a bachelor's degree in computer science with knowledge of software engineering. Ten students had at least some professional software development experience, and several had many years of commercial experience.

The subjects were randomly assigned to one of two conditions – one using TagViewer and one not. We will refer to these as the tagging condition vs. the video-only condition. In both conditions, the subjects were instructed that they would be playing the role of a knowledge acquisition team member. They would first "sit in" on a knowledge acquisition interview and would then perform software development tasks based on the information in that interview several days later.

The KA session was the same video we used in the previous experiments, an hour-long interview of two airline pilots, focusing on take off procedures and checklists with the purpose of designing an electronic checklist system for the cockpit. We tagged the transcript ourselves, attempting to focus on the most important and most likely phrases to tag. We tagged at a conservative rate based on our earlier design studies of TAGGER. A portion of this tagged transcript can be seen in the TagViewer screenshot in Figure 5.4.

Subjects were instructed to take whatever notes they wanted during the session and could use both the notes and the recorded video to perform the software development tasks. Additionally, the subjects in the tagging condition were given a brief training session on the tagging process and told they would have a tagged transcript along with the video to perform the tasks. We did not have these subjects actually do the tagging as this process takes additional training and practice and would likely distract novices from the actual conversation initially. Instead, we wanted subjects to use their natural methods for remembering information.

The second session took place 3 to 7 days following the interview, to simulate some of the forgetting that would naturally occur between when an interview took place and when the knowledge was used. Subjects were told they would be producing a requirements list based on the information that was contained in the KA interview. To give them an initial structure to work within, they were given a simplified requirements framework of five categories, but were told they could use any framework or none at all if desired. The framework was:

- Functional requirement – Describes functions the systems should implement
- Hardware/Software requirement – Describes how the proposed system needs to interact with existing hardware or software systems.
- User interface requirement – Describes an element that the customer wants as part of the user interface, or how the proposed system needs to interact with users.
- Non-functional requirement – Constrains how the functional requirements are implemented.
- Organization Rules / Policies – A rule or policy describes how a work procedure has to happen. These vary from suggested guidelines to mandatory instructions.

Additionally, the subjects were instructed to create as detailed, accurate, and complete a set of requirements as they could in 45 minutes. We constrained the time to encourage efficient use of time and to keep the total session time to an hour.

Finally, we told all subjects they could use their notes, their memory, and the recorded video. In the video-only condition, subjects had a simple video player with standard play/pause and timeline controls for playback. In the tagging condition, the subjects had TagViewer, which contained this same video player in addition to a tagged transcript that indexed the video. The subjects were given brief training on this interface.

We gathered a variety of data to investigate both the performance and behavior of the subjects. To test our hypothesis, we analyzed and rated the resulting requirements documents to look for differences between the two conditions. To evaluate participant behavior, we logged their interactions with either the video player or TagViewer. Additionally, we videotaped subjects to track their focus of attention, whether on their notes, the requirements document, the video, or elsewhere. Finally, we interviewed subjects afterwards about their strategies for creating the documents and their use, or non-use, of the video and TagViewer. We summarize each set of results below and discuss the implications of these results in the following section.

5.5.2 Study Results

The quantitative comparisons in the requirements documents did not yield strong results. However, there were clear differences in the use of video between the two groups. We summarize each set of results below and further discuss these along with the qualitative observations in the following section.

5.5.2.1 Requirements Quality

We wished to compare the quality of the requirements documents created by the study participants. Because of the time constraints and the informal nature of the resulting documents, we began measuring quality through completeness, i.e. quantity of requirements found, and accuracy, i.e. the number of misinterpretations or wrong assumptions introduced.

We created a master list of all possible requirements, which resulted in 78 items.

The participants created a wide variety of requirements documents, ranging from bulleted lists to lengthy paragraphs of explanation. We broke each of these documents into lists of individual items and matched them against the master list. The results are shown in Table 5.3. Overall, participants created from 14 to 28 items, averaging 18. There was no real difference between the two conditions in the number of items identified.

Table 5.3. Requirements quality results. The mean number of requirements and errors identified for each condition.

	# Requirements		# Errors	
	Mean	Stdev	Mean	Stdev
Tagging condition	17.7	3.9	0.7	1.2
Video-only condition	18.3	5.6	1.7	1.6

The distribution of requirements was quite varied. Many items in the master list were identified by only 1 or 2 participants, while only 9 items were identified by 6 or more participants. In other words, there was a lot of variation, both in quantity and focus that seemed to be an aspect of individual differences in memory, style, and focus more than any variable of our study. Given this wide variation, we made no further attempts to rate the completeness of requirements.

We also examined the accuracy of the documents by marking any incorrect items, such as invalid assumptions, misinterpreted details, or conclusions that could not be made based on the video. The number of errors introduced is also reported in Table 5.3. Two tagging participants made errors, while 4 video-only participants did. Overall, the tagging condition averaged 0.67 errors while video-only participants averaged 1.67 errors. There is some evidence ($p=0.12$) of a difference between the conditions in the number of errors made. Thus, while not conclusive, it does appear that using TagViewer may have improved the accuracy of the requirements documents. We will further examine this result in the next section.

5.5.2.2 Behavioral Results

Both the TagViewer software and the stand-alone video player were logged and Table 5.4 summarizes the video playing behaviors. Four TagViewer participants played portions of the video, averaging 11.9 minutes, and clicked on the transcript an average 11 times to navigate to parts of the video. Only 2 participants in the video-only condition used the video, and each played the video for over 10 minutes, averaging about the same amount of play as the tagging condition. However, they played fewer, but longer segments, of the video than the TagViewer users. Additionally, TagViewer users clicked on an average of 10.8 tags in the tag summary window and searched for an additional 1.8 phrases in the transcript.

Table 5.4. Video behavior. For those participants who used video, we list the number of plays, the number of seeks, and the total time the video was played. Seeks for tagging subjects were by clicking on the transcript. Seeks for video-only subjects were by clicking on the video timeline.

	# Plays	# Seeks	Total Video (min)
Tagging subjects			
Subject 2.	10	20	16.2
3.	6	7	3.6
4.	28	13	20.4
5.	4	4	7.5
Mean	12	11	11.9
Video-only subjects			
Subject 3.	3	12	10.1
4.	5	23	12.7
Mean	4	17.5	11.4

We also analyzed the videotapes of the participants, logging where participants were looking as a measure of their focus of attention. This adds to the information from the software logs to tell us when participants were paying attention to TagViewer, reading the tags and transcripts. These results are shown in Table 5.5.

Participants in the tagging condition spent considerably less time looking at their notes and at their requirements document than the video-only condition, and spent that time focused on TagViewer. Additionally, several of the tagging participants had less than one page of notes, which resulted in nearly complete lack of use of their notes, and substantial use of TagViewer.

Table 5.5. Focus of attention. Shown as mean (St. Dev.) in minutes of time spent looking at notes, at the requirements document, and at TagViewer or the video.

	Notes	Document	Video/TagViewer
Tagging condition	2.1 (2.0)	25.1 (9.6)	16.9 (11.5)
Video-only condition	8.7 (5.4)	32.4 (5.2)	2.2 (3.5)

5.5.3 Qualitative Results

While we did not find strong quantitative evidence of our hypothesis on the benefits of capture of access, a deeper examination of both the performance results and the actual use and behavior of the participants provides additional support for our claims, along with a detailed understanding of how tags as indices are used. Below, we further discuss our findings in terms of the potential performance benefits and the general use and perceived value of TagViewer.

5.5.3.1 Performance Discussion

Our study did not show any quantitative differences in the completeness of the requirements produced. The open-ended nature of the task resulted in greater performance variability than we expected. Yet, there are a few interesting things to note. First, TagViewer users did waste at least some time learning to use and utilize a new tool. Additionally, using TagViewer took time away from composing and thinking about the requirements documents. Tagging participants spent an average 7.25 fewer minutes focused on the document. Despite this loss of time, participants identified just as many requirements as those in the video-only case who spent very little time in the video. We had expected video-only participants to waste time inefficiently searching for information. Instead, most of these participants chose not to use the video at all. Not surprisingly, the video-only participants created longer documents, averaging 163 more words, with the 4 shortest documents created in the tagging condition. Yet, while the extra time spent on the requirements document may have yielded longer documents, it did not yield more complete requirements.

Additionally, many of the video-only participants seemed to complete or run out of requirements to add before their time was up. Several spent time editing their requirements, others added unanswered questions or other notes to the document, and one started the video and sat back and watched for almost 10 minutes. Yet, several of the TagViewer participants felt as though they were not done and could continue awhile. If given more time, several expressed that they would have continued to search for and add requirements. Thus, while not more efficient in this particular experiment, the TagViewer participants may have been more complete in the long run, requiring less work later on to find and add additional requirements that were missed. While this potential benefit is difficult to quantify, we do believe that our study still supports our claim that TagViewer will help improve the use of knowledge in requirements analysis and beyond.

Our analysis also revealed that TagViewer may reduce the number of knowledge misrepresentations and inaccuracies, errors that are important to avoid in requirements activities. The behavior of participants suggests reasons for this possible trend. The participants using TagViewer frequently looked for specific pieces of information for additional understanding or clarification. For example, several replayed one segment of conversation concerning a pilot's solution to a problematic checklist procedure. In contrast, the video-only participants only relied on their notes and memory and did not double-check any issues. As they reported in their interviews, they did not feel they would be easily able to find such information. This perhaps led to more requirements that misinterpreted information from the KA session.

Additionally, TagViewer aided participants in finding specific details they may otherwise have left out. For example, one participant played back a segment of the video several times to hear exactly what color and font size the pilots suggested to show accomplished versus unaccomplished items. That participant was then able to write down those details using the pilot's words, which no other participant did. The potential improvement in accuracy through greater elaboration of details is also difficult to quantify in this small study.

5.5.3.2 Tag use and perceived value

This study confirms that even moderate amounts of video and transcripts alone are difficult to utilize. None of the video-only participants found the video particularly useful, even though 2 participants played 10 minutes of video! Only one of these participants successfully found information on one occasion. And this was helped by his foresight to put timestamps as his own personal index in his notes. The other attempted to play the video in the background while he worked, hoping that something interesting would come up on occasion. However, this proved too distracting, and he gave up on the video altogether. Yet, every one of the tagging participants thought that TagViewer was useful or could be. Thus, it was the indexing that allowed the transcript and video to be utilized.

This study shows several ways that these semantically meaningful indices are used. First, the highlighted tags made the transcript easier to browse and scan. Automatically transcribed transcripts, filled with errors, are not always easy to read, and the highlighting of potentially important words or phrases helped summarize and distinguish chunks of text.

The tags themselves were useful even when viewed out of the context of the transcript. All participants spent time scrolling through the tag summary window, often in more than one tag category. Participants stated that tags would catch their eye or jog their memory, reminding them of an issue they had not yet covered. Users also searched through the domain-specific and domain-independent tags lists, looking for particular concepts to help them find desired points in the transcript.

While we expected the conversation tags to be the most useful for this exercise, participants varied in what they used. Two participants scanned the domain specific tags looking for terms they did not know or had not dealt with yet. Several found that the domain independent tags provided more structure to the information, making them easier to use to find specific topics. Additionally, these structures contained details that could go in a requirement, such as checklist item color suggestions. Two participants used the conversation tags, although one of those stated that she discovered the usefulness of these tags way too late, and would likely have started with them had she realized. Several participants also stated that the conversation tags might be good to use to check a completed requirements document, ensuring that the requirements covered all of the tagged issues.

Thus, all of the tag categories were used by participants in this study, in a variety of ways. These multiple strategies confirm our desire to tag from multiple perspectives, providing different structures that are useful for different tasks or preferred by different people. This user-driven indexing method does appear to support knowledge inspection, searching and finding knowledge in its originating KA session. As we saw from the video-only participants, most did not even bother to look in just an hour of video because they thought it would be too difficult or time consuming to find information. In real projects, with many hours of video, this task is even more difficult, and thus, even less likely to occur. Tagging makes this process possible, and potentially easy.

One confound of this study is that we cannot evaluate the value of the transcript versus the tags. In this first study, we focused on comparing the standard of today with our envisioned captured discussion. This question of incremental value deserves follow up study to tease apart the use of the transcripts and of the tags. Our results, however, do demonstrate that the semantically meaningful tags were useful to our participants in a variety of ways. Users navigated the transcript using tags an average 10.8 times. With just a transcript, users could still perform keyword searches, which they did only 1.8 times per session in this study, some of these being variations of the same word or phrase.

5.6 Related Work

In the related work discussion in Section 2, we discuss a number of meeting capture and access prototypes. As with the SAAM exploration, very few explore this type of discussion. The most closely related system would be the WorkSpace Navigator (Ju, 2004) which captures the physical workspace of student design groups, who used the captured materials to prepare reports of their designs for class. Kazman and colleagues also suggested using capture to record requirements gathering discussions. Additionally, their Jabber system (Kazman, 1996) demonstrates using lexical chaining to automatically create a set of concepts of a discussion. We would like to investigate how this and other techniques can be used to automatically generate tags, particularly the domain-specific tags. But we believe that the richness of indexing and kinds of structures that we are exploring are still beyond automated techniques.

Within the knowledge acquisition and requirements domain, our work relates to *pre-requirements traceability*, the ability to understand where a requirement comes from (Gotel, 1994). In addressing pre-traceability, projects such as CREWS (Haumer, 1998) and Amore (Wood, 1994) have also advocated multimedia approaches. The CREWS system allows requirements engineers to integrate rich media – video and images of real world scenario observations, into conceptual current-state models, while Amore provides the ability to link multimedia information to requirements. However, none of these tools were meant to help structure and utilize the voluminous and highly unstructured KA session video and transcripts in the first place. Instead, we are first concentrating on providing meaningful indices to enable locating relevant sources of dialogue when needed. Integrating or linking tags and pieces of KA discussion into other requirements artifacts and tools would be a valuable future exploration to enable even more direct traceability of knowledge.

Our approach also has common goals with design rationale capture. Our conversational tags are based on IBIS (Conklin, 1988), a formal representation of the issues, alternatives, and decisions of a design. However, a common problem with such approaches has been that they require much time and effort on the part of initial designers to create such representations, while providing benefit only to other 'downstream' users. Tagging may well suffer from similar problems, but we are taking several related approaches to mitigating these problems. First, we have focused on a simple set of tags to reduce the effort required to annotate the KA sessions. Second, as we have shown, tagging can provide benefit immediately to knowledge acquisition team members and requirements engineers, making it more worth their effort. Finally, rather than placing all of the rationale into one representation, we are instead exploiting concepts already important to software developers to help structure and annotate the rich KA recording, thus making the knowledge and rationale contained within that recording more accessible.

5.7 Summary of Contributions

We have applied automated capture and access technology towards improving the traceability of knowledge learned in Knowledge Acquisition sessions. We have shown that the set of indices can be related to the work of the users, even in the case where artifacts do not yet exist. In this case, the indices are instead structured around future work. Additionally, the indices we used are more than simple bookmarks, the semantics of the tags allowed them to be used in a variety of ways to remember and find information. Additionally, we examined a different balance between cost and benefits. Most meeting capture prototypes have attempted to keep costs low by not requiring changes in user behavior in the meeting, an approach we took in both TeamSpace and SAAM. Yet the capture needs in many of these situations are also often low, as we found in our TeamSpace studies. Thus the benefits from many of these systems have not been realized. While the effort to produce these proposed tags is high, we believe that the needs and potential benefits are also great, and potentially worth that cost.

While still early, we believe we have successfully demonstrated the benefits capture can provide to these types of important discussions. We have shown that the use of captured KA sessions does have a clear affect on whether the video recordings of those sessions are utilized. The TagViewer prototype allowed participants to use semantically meaningful indices to browse, search, and find relevant information in the transcript and video, while those without these capabilities did not take advantage of the video. Additionally, we found limited, yet positive evidence that this added support for knowledge inspection can improve requirements related activities. We now need to examine this method in a larger context to understand how capture and access supports knowledge traceability within real, existing knowledge acquisition and software development processes. We believe that the capture and indexing of many hours of video will help provide access to that important base of knowledge, and will be even more beneficial on a larger scale. We also must evaluate the balance between costs and benefits of our capture system. Our early results encourage us to continue to investigate whether the effort involved in TAGGER is reasonable in providing useful access to recorded knowledge acquisition sessions.

CHAPTER VI

DISCUSSION AND CONCLUSIONS

In this thesis, we have explored a variety of issues in meeting capture and access by designing and evaluating several different prototypes. With SAAMPad and SAAMPlayer we demonstrated the application of capture and access services to the software development domain, and showed that capture could be structured around a very specific domain artifact. With TAGGER and TagViewer we explored the concept of user-driven semantic indexing, which is more feasible in such a high-need domain. We also demonstrated the use and benefits of capture services on a realistic, complicated task. With TeamSpace, we deployed a robust, scalable, and usable system where we could understand detailed access behaviors and long-term use and implications. In each exploration, our evaluations have focused in part on how users can search, browse, and find information in the recorded meetings. These navigation behaviors were supported by a variety of different indices in each application. While we sought to understand the benefits of capture in each situation, we also examined the costs and barriers to using capture and access services. With our experiences and results, we suggest a number of design tradeoffs and guidelines for future explorations of meeting capture and access. Finally, while this research has advanced our understanding of the use of captured meeting information, there are many ways to further this research and explore still unsolved problems.

6.1 Review behavior

With each of our access prototypes, users navigated the meeting record in different ways:

- In SAAMPlayer, subjects used the architectural components and key issues in a summary document of the meeting to indicate potential segments of interest on a timeline.
- In TeamSpace, users reviewed meeting artifacts and used those artifacts and a timeline to navigate to portions of the audio. Their navigation behaviors included scan, jump, skim, hone, random, and replay.
- In TagViewer, a complete transcript indexed the video of the meeting. Tags served as indices to locate areas of interest within the transcript.

With the TeamSpace evaluation, we identified six low-level navigation behaviors that users exhibited in searching for information. In evaluating SAAMPlayer and TagViewer, we observed some of these behaviors being used, and also identified some differences in the review of the meeting content. In SAAMPlayer, users navigated using a summary document of the meeting, prepared after the fact. Yet, the jump behavior was not supported with this artifact. Each component instead related to a number of points in the meeting. Thus, users instead had to skim to find information. The keywords-as-indices merely helped them narrow down the places to perform skimming. As in TeamSpace, users requested interface support for their skimming behavior. They also requested additional indications of the structure of the discussion to aid their review.

With TagViewer, users had the transcript that they used to navigate the video. Thus, they performed no skimming or honing whatsoever. Instead, they had to scan the transcript, which does appear to be a quicker activity than skimming audio or video. And rather than randomly moving to a point in the video, they occasionally randomly scrolled in the transcript. This behavior provides an additional definition and understanding of scanning beyond the scanning performed in TeamSpace. Yet the transcript alone was not sufficient. Most users started a search with the tags and used them to index into the transcript. Without these additional indices, users would have had to rely on their scanning and keyword searches. Additionally, the highlighted tags in the transcript aided scanning by allowing users to scan tags and their surrounding text instead of large, unmarked chunks of text. Yet, as we saw in SAAMPlayer, important keywords will not relate to any one point but potentially hundreds. Thus, while better than skimming, scanning will not be a trivial activity with many hours of meeting content and still needs to be supported with additional structure and indices.

In the study of the Tivoli user, Moran and colleagues observed how the user's behavior evolved from mostly playing audio straight through to more focused, non-sequential replay (Moran, 1997). In his evaluation of eClass use (Brotherton, 2001), Brotherton expands upon this to characterize the review behaviors of lectures as StraightThrough, StopStart, Skip Ahead, Relisten and NonSequential. He found that 62% of the media access was StraightThrough and only 19% was NonSequential. However, in all of our evaluations, including the deployments of TeamSpace, almost all of our users' behavior would be characterized as NonSequential. One difference is that meetings are less structured than a lecture, and we gave our subjects multiple questions to answer that were contained within different portions of the meeting. However, the eClass result may also be partially due to interaction issues, such as the audio streaming delay and the lack of an integrated timeline that discouraged some navigation. Thus, this kind of characterization is not sufficient to understand and support the user behavior we observed.

Moran also coined the term “salvaging” to describe “the new activity of working with captured records” (Moran, 1997). Basically, salvaging consists of culling through the meeting for useful pieces of information, relating them, organizing them, and producing new material with them. In most of our evaluations, users were searching for the answer to a question. Thus, we would not characterize this as salvaging, but as more a searching task which will be a piece of any salvaging activity. In the more open-ended task in the TagViewer use, several users did start to engage in salvaging, but not enough that we can discuss support for that behavior in detail. Thus, we believe that there are still other behaviors and interface issues to explore that support users going through meeting content to pull out, organize, and further work with important information.

6.2 Indexing

The navigation behaviors were supported (or not) by a variety of different indices in each application. Perhaps the easiest lesson to draw from the three prototypes is that any and all indices are useful and desired. In Chapter 1, we introduced the notion of explicit indices, indices that are created based on interaction with the capture and access services. We have demonstrated the use of this approach in each of the prototypes, which allowed us to create a variety of meaningful indices in several different domains.

- In TeamSpace we captured the session-related activity of people joining and leaving the meeting, both automatically and manually.
- In both TeamSpace and SAAMPad we captured indices based on the interaction with domain artifacts. In the case of TeamSpace, some of these artifacts are based on the meeting itself – the agenda and the notes of the meeting. Others are general, common artifacts in the workplace – action items and presentations. In

SAAMPad, the indices are based upon interaction with a software architecture diagram. The interactions we support range from creating and viewing, such as with action items and software architectural elements, to editing and annotating, as with agenda items and adding notes to a slide. Additionally, we supported interactions specific to the artifact – checking off an agenda item and manipulating the architecture diagram.

- We examined user-driven indexing in the knowledge acquisition domain where no artifacts generally exist. While this manual method is unusual to consider in a ubiquitous computing environment, we demonstrated its feasibility in design prototypes. Even though this would certainly change users behavior, we did not change the overall activity – capturing the discussion.

6.2.1 Content-based indexing

Throughout our prototypes, we have focused on content-related indices that somehow give indication of the structure of the meeting or the topic of the discussion. Session-related activities or contextual indices are generally non-persistent and, most important, simply mark *when* an activity was started or an event occurred (e.g. when the meeting started, when someone left early). Indices based on interaction with artifacts are semantically more powerful because they describe *what* people are doing with objects from their work environment and relate to the content of the discussion. Additionally, those artifacts may contain useful content themselves, not requiring actual replay to gain benefit.

We can begin to see the implication for replay when users have very fine-grained indices based on content. Not a single subject using TagViewer used the timeline to navigate or replay the video. Instead, they naturally used the transcript for all navigation. In TeamSpace, the majority of the navigations occurred using the timeline, where long segments had to be searched without any other content-based index to use. While users of SAAMPlayer interacted with the timeline to navigate, they chose where to navigate based on a hot spot or overlapping area of desired keywords. Thus, while users still skimmed, their skimming was directed by the content -- those keyword locations.

However, it is not just the details of the discussion content, but the structure of that discussion that is important. In TeamSpace users often started with the agenda, jumping to the topic of choice before consulting the notes, which sometimes contained more details. Many participants in the MeetingViewer study also requested more detailed agendas with subtopics to provide a more fine-grained structure. The transcript in TagViewer indexed the video by exact content – by the words of the discussion itself. Many TeamSpace users also request such a transcript. However, simply adding such a transcript is not sufficient. In TagViewer, users more frequently started with the lists of tags, a set of semantically meaningful indices that provided detailed organization and structure of the discussion. While a text transcript may be easier to scan than audio is to skim, scanning hours of transcripts is still tedious and difficult, especially with many inevitable recognition errors. Not only did the tag lists provide an easy structure to view, highlighting the tags in the transcript provided visual structure to the transcript text, making it easier to read and scan. Additionally, as is seen in the density of the keywords locations painted on the SAAMPlayer timeline, important words or topics may be repeated many times throughout a discussion. Users requested additional structure to the timeline to show the various segments of the discussion. Thus, while keywords can help indicate discussion topics, some amount of interpreting and searching those locations may still be required if additional structure is not provided, especially with many hours of discussion.

6.2.2 Context-based indexing

While content or semantic indexing was most valued by our users, contextual cues are still important and useful in finding and retrieving meeting information. We only provided people join and leave events in TeamSpace, and these events were occasionally used to narrow a search based on when such an event occurred. Additionally, many participants in the MeetingViewer study expressed memories based on such contextual cues as who was speaking or who was in the room, and suggested adding indices such as speaker change or speaker identification. Such indices would aid in quickly skimming the audio or video, narrowing down which portions to replay. Additionally, they can be used to infer structure or content when little is provided, such as determining what the topic of discussion was based on the speaker, or distinguishing between a presentation with one speaker and a discussion with many interleaved speakers. These additional indices would also help distinguish real topic discussions from side discussions or informal chatting – segments which will not likely be marked on any agenda or other formal structure.

Capturing context is important, and a highly researched topic in ubiquitous computing and multimedia. As we discussed in Section 2.2.2, many of the prototypes that derive indices are capturing context-related indices rather than content-related ones. Much progress is being made in this area. Yet we believe too little focus has been given on successfully and reasonably capturing more explicit artifact-related indices. We have demonstrated that these content-related indices can be captured in a reasonable manner, even when done explicitly, and should be explored more frequently.

6.2.3 Unexplored indexing

There are other forms of indexing we have not explored that our evaluations suggest would be useful. For example, all of our indices were captured in the meeting. There are a number of indices that could be created based on access behavior, such as users marking areas of interest and providing additional structure, notes or keywords. This would be useful to support the "scavenging" behavior Moran describe as users continue to work with and organize discussion content over time. We would expect that any real users of the TAGGER system to display this type of behavior as they continue to organize and work with knowledge in a variety of development tasks. Post-hoc indexing could also be done automatically based on user interaction with the access interface, saving and visualizing a history of the user's interactions, or perhaps other users' interactions. For example, in SAAMPlayer, and even occasionally in MeetingViewer, users got lost or just skipped too far and unintentionally replayed portions of the meeting they had already visited. Knowing where you are and have been will reduce these incidents. Also, if the captured material will be used by a number of collaborating people, understanding both the spots no one has viewed, and the spots many have viewed, may aid and direct future review.

An important aspect to consider in making meeting material useful is being able to use future artifacts to create indices back into the meeting record. For example, in creating a requirements document based on knowledge acquisition discussions, users should be able to link created requirements to the portions of the discussion they relate to and that were used in their creation. This will allow the discussions to be used by those who rely on the outcomes of those discussions down the road, such as a designer being able to trace from requirements back to related portions of the knowledge acquisition sessions. The SAAMPlayer study showed how the captured discussion can augment such artifacts, as users found the video provided different abstractions, additional details, and validation of the material contained in the summary document. The access to this video was coupled with the topics in the document, allowing relatively easy replay of the video to gain this additional information when needed.

6.2.4 Cost tradeoff

A major lesson to draw from all of these prototypes is that while any and all indices may help review, capturing those indices will require different levels of effort or instrumentation. In the case of knowledge acquisition, the interview sessions contain relatively few artifacts to capture and use as indices. Thus, we relied on an effort-intensive, manual indexing method. This method would certainly not have been acceptable in TeamSpace, where we aimed for the exact opposite – as little effort as possible. That low effort was still often too high for our users and other have investigated completely passive capture systems with derived indices (e.g. Schultz, 2001). In fact, while all of the indices TeamSpace captures greatly helped review, several users suggested the desire for even quicker, cheaper capture that was always available, perhaps on a PDA. With such a system, we would reduce the number of artifacts we could support, recording fewer indices and making review potentially more difficult. However, as authentic users of TeamSpace often replayed large portions of audio, this may not negatively impact some of the real use. There will likely exist several of these "sweet spots" where the cost of capture is balanced against the cost of accessing information in different ways.

6.3 Cost versus Benefit

As with any application, users must deem the perceived benefits of a capture and access system worth the cost of learning and using the system. The challenge in ubiquitous computing is that people have difficulty envisioning their use of such new technology. That said, the feedback from all prototypes was very positive. In fact, the feedback from the Ubicomp group on their use of TeamSpace seemed to exceed the actual use we saw in the log files. Thus, in many situations, users at least see potential benefits and are willing to try using such services. We have demonstrated some of these benefits in our various evaluations. And yet, the question of whether these benefits are worth the costs is still mostly unanswered. With SAAM and TAGGER, we did not measure or evaluate any real costs or barriers to use, and can still only speculate the real worth. With the deployments of TeamSpace, we did observe real use. However, that use was infrequent and not encouraging. In examining this question of cost, we first discuss the various costs involved in capture and access, better characterized as the barriers to use. We then continue the discussion of costs versus benefit in comparing low and high need situations and argue that capture may only be worthwhile in high need situations.

6.3.1 Capture and Access Barriers

Despite users openness to the technology, there are still many barriers to using meeting capture and access services, especially in everyday situations. Many of these barriers are similar to the deployment of any interactive application, so we will focus our discussion on examples we observed in the context of our capture and access applications.

First, the users need to know and remember that the services are there when needed or wanted. We were able to influence when capture occurred in TeamSpace with several motivated users. However, very few of the remaining users thought to use capture in other meetings or discussions. They said that it had not really occurred to them that they could do so. Additionally, during the access behavior study, many commented that they should take advantage of TeamSpace more, but normally do not think about it. Thus, the access behavior study served as a way to bring TeamSpace to the foreground and remind users that the meeting content was even there. Additionally, users need to remember what the system can provide them. Despite sitting next to the scribe in the meetings and observing her typing notes, several Ubicomp group members initially forgot about these notes in the access behavior study. Many users never interacted with MeetingClient and were not sure how the indices were recorded and exactly what they represented. While the capture required very little interaction by most users, the capture occurred mostly in the background of user's attention, in turn limiting further use of TeamSpace and the meeting record. In contrast, the author was very much aware of the system and the resulting content, and thus reviewed that content more often.

In any deployment of such a capture and access system, we will need to explore ways to jumpstart the use, and provide occasional reminders that meeting records exist and can be accessed so that users can begin to incorporate capture and access into the rest of their activities. In TeamSpace, the system emailed action items after the meetings, which often served as a reminder of the content as well. However, the Ubicomp group used action items relatively infrequently. Those and other kinds of reminders will be especially important for users who did not participate in the discussions themselves, such as designers who may need to refer back to knowledge acquisition sessions well into systems development. Integrating access into other tools or applications will also serve this purpose, as users are reminded of related content and can access content in multiple ways.

Other capture and access barriers relate to the costs of using the actual applications and interfaces. Users have to first understand how to start capture and access services. With TeamSpace, this meant finding the web site and remembering how to navigate through it to create or find a meeting. Users also had to have components or libraries installed on their machine in order to either capture or review a meeting. This prevented many people from glancing at meetings out of curiosity, or from reviewing a meeting on a new computer. Once started, users must then be able to find and use the features of the various applications, especially challenging for infrequent use. For example, users had difficulty remembering the functionality of MeetingViewer between uses, especially how to navigate using agenda items or slides. During the experimental studies of both TagViewer and MeetingViewer, subjects could, and often did, ask the expert running the experiment how to perform some action. However, this quick expert help was not available in real use. In many cases, whether trying to start MeetingViewer, play the audio, or perform some other action, if the first attempt did not work, users often gave up. A challenge in deploying research prototypes is that help capabilities are likely to be limited. For example, MeetingViewer itself had no help system; the help files were all on the main TeamSpace web pages and many users did not realize that they existed.

Users will not just have to know how to use the features of the applications, they will need to have good strategies for using those features to achieve whatever goals they have. These strategies will take time to form and will vary among users. As we saw in the access behavior study with MeetingViewer, several users defaulted to skimming through audio instead of looking for other indices to narrow their searches, which resulted in wasted time. Different user strategies were particularly evident with the use of TagViewer. We expected users to mostly use the conversational tags as a start at their requirements lists. However, none thought to use the tags in this fashion, and one subject stated that she discovered the usefulness of these tags too late and would have liked to utilize them more for that purpose. Several liked the domain dependent tags for the structure they provided. Yet others preferred the domain independent tags as unique keywords that served as memory aids or topic identifiers. Two subjects spent a significant amount of time reviewing the recorded KA session, writing requirements as they worked their way somewhat linearly through the discussion. Others only used TagViewer to look up details or verify information. Thus, even with just 6 users of TagViewer, we saw a variety of different and successful strategies emerge. Capture and access services need to suggest and support reasonable strategies initially, as well as allow the creation and customization of personal strategies over time. For example, in the work with Tivoli (Moran, 1997), the user who reviewed the meetings eventually created his own special symbol to indicate portions of the meeting he wished to revisit. These types of unique strategies will develop only after regular use of capture and access services.

The final barrier to overcome in meeting capture and access is our comfort and experience in current methods. We have all been having discussions and remembering them all our lives. We have used personal notes and other memory aids since grade school. Even when people have notes, many still rely heavily on their memory because that is often sufficient. We are also used to our memories failing, and dealing gracefully with such failures, such as asking other people or just making a reasonable guess. It will be hard to change such ingrained processes. We observed this difficulty in the TagViewer study, where two of the experts with many years of experience in requirements and software engineering relied heavily on their notes and their own existing practices of creating such documents. Both consulted TagViewer toward the end of their sessions, but neither replayed any video or used the tags and transcript to the extent that the others did. Replacing or augmenting such expert behavior will take time and may encounter resistance.

Additionally, in most groups and environments, existing tools are already used and adopted, such as calendars, group workspaces, document applications, and others. These tools are useful above and beyond meetings, and requiring a change to this practice or repeated work just for a meeting capture and access system will likely not succeed. Thus, capture and access services should be integrated with these existing tools to reduce these barriers. However, there are many different tools and applications for common practices that are adopted differently by organizations and groups. And integrating with the wrong components will limit the use, as occurred with TeamSpace. We attempted integration with a group workspace that was not needed by our real users. This integration then became an added barrier. These issues will make such integration challenging for developers of capture systems.

6.3.2 Low versus High Need

In this research, we have explored different domains with different capture needs, and thus different cost and benefit balances. While we wanted to deploy TeamSpace to higher-need groups, such as one of the groups at Boeing, our actual deployments were also in low need situations. The meeting artifacts did not contain many details, and individuals rarely needed to review any content. There were few known outcomes or uses for the information in the meeting. As we discussed in Chapter 4, meeting capture and access can still provide many benefits in these situations, and our users were very positive about the potential. However, smaller benefits naturally means that users will require lower costs or barriers, perhaps impossibly low. If the meeting capture were not essentially free for the Ubicomp group, they likely would not continue their use of TeamSpace despite their preference for having the records available whenever they want them. Additionally, in everyday types of discussions, there may actually be *more* tools and techniques already in use. Thus, integrating capture with the rest of the work environment in order to reduce some of the barriers may actually be more difficult than in more unique domains.

We consider the architectural analysis sessions and the knowledge acquisition sessions high-need situations. The information discussed in these domains is important to remember, and documents summarizing or using that information are often prepared and consulted throughout further development. Some effort is often already put into documenting such discussion details, whether with a dedicated minute-taker or videotapes. These practices show that the information is worth some cost to preserve. Yet, there are still problems with existing practices in losing important details, or having details that are too difficult to find and review. Thus, capture and access may be particularly appropriate in these situations, providing improved access to recorded information where users are willing to pay some cost to achieve that goal.

However, high-need but rare situations may suffer from some of the same problems as low-need situations. The SAAM analysis meetings are generally held only once during the lifetime of a system. While remembering the information may be invaluable, a group is not going to adopt an entirely new system just to document that one important meeting. Reviewing such a meeting, no matter how important, will share many of the barriers of the low need situations. Users will not be aware that the capture and access capabilities exist; they will be unfamiliar with the system software and interfaces. It was this problem that led us to move from the SAAM work to the very general, and potentially every day, TeamSpace system. Yet TeamSpace did not provide enough benefits to our groups of users. We believe we finally found the right balance of high needs and high benefits in the knowledge acquisition exploration. Yet we must still explore the real costs of using a system like TAGGER before we can make such a conclusion.

One of the problems with much of the existing research in capture and access is that by focusing on generic meetings as TeamSpace does, the systems are also addressing mostly low need situations. Relatively few systems are attempting to address more focused, high need discussions. The exceptions are similar in spirit to our SAAM and Tagger projects, Workspace Navigator (Ju, 2004) and the Design Meeting Agent (Hammond, 2002) capture different kinds of design meetings. And Workspace Navigator was useful to several groups of students in preparing their design reports with captured material. With such lack of focus on understanding and addressing real information problems, meeting capture research has thus far not lived up to its potential to benefit the workplace and does not appear to be making much progress in doing so.

6.3.3 Additional Challenges

Capture systems could be considered a form of groupware, and thus share many of the same implementation and adoption issues of any groupware system (Grudin, 1994). For example, the benefits of each recorded meeting may not be shared by a number of people. TeamSpace provided the most benefit to an individual who missed the meeting, or who received feedback on an idea. These were infrequent occurrences for any one person. Thus, use remained infrequent and required a long period of time to gain enough use to evaluate. We were able to do this to some extent with TeamSpace thanks to several dedicated capturers, but long-term deployments of a such systems remain challenging and costly.

Capture systems also share some of the same goals as design rationale capture techniques. One of the problems that the design rationale capture community faced in the 1980s is that those who were responsible for capturing the information were not those who would ultimately benefit from that information (also a groupware issue). Users will be unlikely to exert even a small amount of effort to capture a discussion if they do not see any benefits for themselves. Thus, meeting capture services will be easiest to introduce to people who can immediately benefit from them in some way, even if perhaps more important benefits may not be realized for some time. This is the case with the knowledge acquisition sessions, where the information within the session is immediately useful and necessary. While the ultimate goal is to facilitate knowledge traceability throughout the use of that knowledge, those in the knowledge acquisition and requirements phases can use the captured sessions to find and use detailed knowledge more easily. We demonstrated how meeting review can be used in such an immediate task. We will only understand the use of captured KA sessions by later designers and developers if capture is first adopted by requirements engineers.

Thus, despite many intuitions and evaluations, including our own, suggesting that capture and access is a worthwhile activity, we have yet to really demonstrate that worth. We do believe that our contributions help to advance our understanding of those potential benefits and barriers, and encourage future explorations of potentially more worthwhile, high need situations.

6.4 Other issues

There are a number important issues in capture and access that we did not explicitly study, yet have a few comments on. One is the effect of capture on the meeting behavior itself. In Chapter 4, we discussed some feedback with TeamSpace users that implied that the recording did not have a big impact on user behavior in the meeting. Anecdotally, several users reported that they took fewer notes than they would have otherwise because they knew TeamSpace was recording the meeting. While this self-reported behavior does imply a certain acceptance and reliance on the system, we cannot report whether any real and measurable difference occurred or whether this is a positive outcome. Additionally, we were not able to deploy TAGGER to study whether the reduced note taking and meeting participation that would occur by those who must perform tagging would negatively impact knowledge acquisition. Surprisingly, few users expressed any real privacy or other concerns with recording their discussions. Most were comfortable with recording and stated they did not change what they said. From this, we do not conclude that there are no concerns or changes that occurred, but that instead our evaluations, user groups, and environments were not sufficient to study this important issue.

We did observe an interesting phenomenon in the access behavior study of MeetingViewer. Participants sometimes thought they knew an answer, but decided to review the meeting anyway and make sure. Perhaps this act of committing to an answer in writing led them to not trust their memories and instead spend time verifying the answer in the meeting record. They felt more accountability for their answers than if they had just been asked the question by a colleague. This implies that capture and access may be most useful in situations where content from the discussion is written down or used in written artifacts.

Another issue of accountability was raised by multiple users throughout our evaluations. Recording discussions makes people in those discussions more accountable for what they say, since an indisputable record will remain. Those who use such discussions will also be more accountable to accurately reflect the details in the discussion. In general, those we interviewed considered this added accountability positive. First, in domains such as knowledge acquisition, it is important to be able to use the words and details of the users. For example, in the KA session we used in our studies, the pilots suggested the colors “amber” and “magenta,” not red and green. Presumably, these colors have some meaning or regular use for the pilots. Using all of these rich details is challenging without access to the original discussion. Additionally, several of those interviewed suggested that recording would keep people on target and less likely to discuss tangential, unimportant things. This surprised us as we would generally consider the suppression of certain topics of conversation a negative outcome. Other drawbacks to this potential accountability may be that the details of an evolving discussion could be taken out of context and used when they are no longer valid. Or, personal statements could be used against an individual for many reasons. However, few of those that we interviewed expressed these concerns. The real impact of this accountability will only be understood with long-term evaluations in real work settings. Perhaps the real price and benefits of added accountability will not be revealed until a larger volume of material is captured and some of those negative effects can occur.

6.5 Design guidelines

The lessons we have drawn in designing and evaluating our research prototypes can aid others in designing new meeting capture and access services. The most basic lesson is to first consider the goals of the user and their use of the information. This is an obvious design tenet. However, many of the meeting capture and access systems built have paid more attention to capture issues and neglected consideration of the access goals of users. We found relatively few issues with the capture technology that impacted real use, but instead found issues with access and overall barriers to use such applications. We summarize all of our results and discussions in several design tradeoffs and a number of design guidelines to aid future research of meeting capture and access.

6.5.1 Design tradeoffs

Our explorations have uncovered a number of possible tradeoffs to consider, which we have discussed throughout this thesis. First, there is a tradeoff between tightly integrating capture and access with other tools or processes in order to better relate to the work context, and in supporting a wider range of users. Our integration of capture and access in a group workspace in TeamSpace may have helped distributed teams, yet hurt all of our co-located teams.

A second tradeoff is between making capture as invisible to the users as possible, and in keeping capture and access in the foreground so that users will remember and be encouraged to use the services. Keeping capture in the background lowers capture costs, possibly leading to more content being captured. Yet, if users are mostly unaware of that content, access of that content will be less likely and thus less useful.

Finally, one of the most important tradeoffs is in capturing the indices. The more indices captured, the easier access will be. Yet, capturing these indices, especially those that are semantically meaningful, raises the barrier to capture and again may hurt repeated use.

There will be different "sweet spots" of these tradeoffs, even in the same domain. For example, users of TeamSpace suggested that perhaps they would capture more their individual meetings if capture were more readily available on a mobile device. With more content, access may become frequent enough to become regular practice. Yet this would also lead to even fewer indices being recorded, making it difficult to access information within the meeting records.

6.5.2 General guidelines

Many of these design guidelines reflect our successful approaches and summarize earlier evaluation results and discussion and also reiterate the above tradeoffs. In designing meeting capture and access services that support users' information needs, there are a number of capabilities to consider:

- Capture as many indices as possible for the allowable costs. Consider the tradeoffs in costs to the user and the benefits provided.
- Indices that provide a semantic structure to the recording, and link to specific

content are the most important to provide.

- Capture the artifacts users are working with, including documents, notes, or informal documents. Structure the recording around these artifacts.
- Integrate capture and access with existing artifacts and tools as much as possible. However, consider the tradeoff of added cost if that integration is not right for a group of users.
- Support the entire meeting process including preparation and meeting outcomes, not just the recording.
- Allow for easy expansion and evolution of the capture and access capabilities, such as the addition of new artifacts or indices.
- Provide easy launching of the capture and access application, possibly even automatically starting.
- Provide frequent reminders to the user of the capabilities and availability of the capture and access services.
- Consider portable solutions or systems with few technology constraints that can be easily used in many environments and situations.

6.5.3 Capture guidelines

While much of this research has focused on the access behaviors and interfaces to support them, we have also learned some lessons on designing and implementing the capture software.

- Support collaborative processes and a variety of inputs to gather input from as many users as possible.
- Support, do not impose, the structure of the activity when possible.

- Support personal notes or annotations.
- Allow external artifacts to be imported or captured. Allow annotations and notes to be exported for other use.
- Record reasonable audio quality. High quality audio is not necessary unless the audio stream is to be automatically analyzed.
- Video is less important for most activities than the audio. Video will likely be more useful if there is considerable movement, or if certain artifacts or interactions can only be adequately captured with video.
- Allow capture to be quickly paused, or muted, so that sensitive discussions can occur without recording.
- Provide visual cues that signal recording is occurring. These cues can be something as simple as a microphone on the table.
- Support graceful degradation when problems or errors occur. This means allowing the capture to continue even when some part of the environment is having problems. For example, in systems that rely on networks, create local, backup copies of information in the event of network loss so that no captured information is ever lost.

6.5.4 Access guidelines

As we continue to state, the most important aspect in designing an access interface to consider is what the high level goals of the user are, and how those will break into lower level patterns of behavior. We have identified a number of low-level behaviors that support browsing and searching and observed different behaviors in the different prototypes. We reiterate the most important guidelines that support these types of behaviors.

- Provide random access with fine-grained control. This is important both to provide access to any desired portion of the recording, but also to allow any unusual or unconsidered behaviors.
- Provide easy navigation based on the captured artifacts. Make this navigation capability obvious to the user.
- Allow the user to quickly scan the available information. Provide visual summaries of the events or artifacts that are available. Also provide easy scanning and navigation of long artifacts. For example, provide thumbnails of the pages and color to call out important areas of text or added annotations.
- Provide fast forward and reverse buttons that move a pre-set amount for quick skimming and replay.
- Provide audio compression for faster replay of audio.
- Provide some sort of help or wizard on the interface that explains the interface capabilities and the meaning of the various artifacts and indices.
- Allow some form of post-hoc indexing, either with notes, bookmarks, or automatically tracking where the user has visited.
- The audio or video needs to be responsive. Long streaming times will frustrate the user. Additionally, the user should be informed of problems or delays in downloading or streaming audio and video.

6.6 Future work

There are still many challenges in automating the capture, recognition, and retrieval of activities that are being addressed within the ubiquitous computing and multimedia research communities. However, we believe that a large challenge still lies in understanding the use of those recorded activities: the goals of the users, and how to support those goals in an application. There are a number of open questions that remain as to the impact of capture and access services on meetings and the workplace. Where and when will recorded information be useful? Where will we be satisfied with existing methods? What privacy and accountability issues remain? What happens as the group evolves over time? How long will users want meeting records stored? How long can such information be useful in a particular domain? Are there ways to remove the least important information so that we can maintain privacy, yet keep the most important information that will benefit users? How can we maintain useful archives of a large amount of discussion recordings? We advocate investing more time in high need situations where the benefits of introducing meeting capture services will be greater and possibly easier to achieve. We will not understand the impact of meeting capture and access in the workplace until we can provide reliable, usable systems that are incorporated into everyday work practices over a long time period in a number of domains.

An important issue in creating better access services is scaling up to more and longer discussions. While TeamSpace was scalable to a large number of groups and meetings, we still have not investigated how to facilitate the review of larger, interrelated discussions. With TeamSpace, the user generally only wanted to review a portion of one meeting. As the Ubicomp group only had one meeting a week, locating this meeting was relatively easy, even after several years. Users may wish to be able to review multiple meetings, or may need to search an archive to find the one discussion they wish to replay in detail. TagViewer was only built to display one hour of video and transcript. We can reasonably expect that requirements engineers would have hours and hours of knowledge acquisition video, with hundreds or thousands of tags. We will need to investigate interfaces and visualizations that support reviewing this large amount of multimedia information.

We have advocated integrating capture and access into the user's work context. This means potentially integrating services into existing tools. In this thesis, this integration did not occur. All our applications were original and stand-alone. And while TeamSpace is an instance of a larger group workspace, that workspace was created as we were creating the capture and access features. Integrating with existing applications is challenging, however, requiring APIs or scripting capabilities to add new functionality.

There will continue to be a tension in providing easy, quick, and general discussion capture with potentially higher access costs and fewer benefits, and capture and access that is customized with the artifacts, tools, and specific uses of the domain. Ideally, we would like to provide a lightweight general capture and access system that can be used in a variety of meetings and customized with the specific artifacts, activities, and goals of different domains and discussions, and that can be easily integrated with a variety of existing tools. This could potentially provide both the regular use and higher benefits we seek. Yet the HCI and ubiquitous computing community are still a ways from such a vision. We first must understand the general features that will support regular common use and how to add and customize those features to better support a particular user group or activity. We must continue to investigate capture and access on each end of the spectrum in order to move towards this possible vision.

6.7 Conclusion

Our goal in this thesis was to advance our understanding of how meeting capture and access services impact work practices through prototyping and evaluating several meeting capture and access systems. We have successfully designed, implemented, and evaluated multiple prototypes in different domains. We have demonstrated supporting the specific activities and artifacts of different discussions, explicitly capturing the interactions by the users to serve as indices into the meeting record. Our evaluations have demonstrated how users can navigate and review meeting records using these indices. Indices linked to the discussion content are the most important to aid meeting navigation, but context-related indices are still vital to support navigation, especially when content-related indices are unavailable. We have observed and discussed a number of navigational behaviors, and the interface mechanisms that can support them in access interfaces.

In investigating the impact of capture and access, we have shown a number of potential benefits for using meeting records. Users valued reviewing action items, missed meetings, and the specifics of detailed feedback on an idea or document. We have shown that recorded meetings provide additional details, different levels of abstraction, and additional authority over documents summarizing a discussion. And finally, we have demonstrated how indices are needed in order to make use of recorded video, and how the captured and indexed transcripts and video may aid future tasks by providing access to forgotten or specific details.

Finally, we have examined the real motivations of using a capture and access system. While there are benefits that users want and value, barriers to use are still a big issue. When and how to make meeting capture worthwhile is still unknown. Our experiences lead us to believe that creating generic meeting systems without a particular problem or outcome in mind will lead to systems that are mostly used in low need situations with few benefits. We will continue to examine capture and access services in more focused situations where we can hopefully address real information problems and understand real use issues with this ever more ubiquitous technology.

APPENDIX A

STUDY MATERIALS

SAAMPlayer evaluation, tasks for participants of original SAAM session

Subject:

Date:

Background
You participated in a SAAM analysis with a group of designers at Company A. You determined the overall architecture of the ATX system, as well as investigated possible changes to the system. Those analysis sessions were videotaped and a summary document was prepared. The document summarizes both the architecture of ATX and the discussion of proposed changes. Take a few minutes to look through the document to review ATX and the proposed changes.

Tasks
You are now moving on to the next phase of the ATX development. A new group of designers is looking to you for direction on system reconfiguration. Imagine you are being asked to provide information about two possible changes that were previously discussed during the SAAM analysis. Please answer the following questions for these new designers with as much technical detail as you can. You can use all of the ATX documentation and the video as much or as little as you wish. When looking for something in the video, please think-aloud what you are doing and looking for.

Task 1: The separation of the midnight rollover from the system reconfiguration shutdown is being adopted. The changes that the SAAM analysis group decided upon for the midnight rollover are being completed. The next step will be to make the changes to handle the new reconfiguration shutdown that doesn't occur at midnight. What will have to be handled differently during the reconfiguration shutdown than in the old midnight shutdown? What kinds of changes will need to be implemented? Estimate how difficult these changes are.

Task 2: A customer has requested that the ATX system not shut down and not lose data during reconfiguration changes. What happens on a configuration change that makes this difficult? What are the issues or problems involved in changing the system to not shut down during reconfiguration changes?

SAAMPlayer evaluation, tasks for non-participants

Subject:

Date:

Background

A group of designers at Company A participated in an architecture analysis of their ATX system. They determined the overall architecture of the ATX system, as well as investigated possible changes to the system. Those analysis sessions were videotaped and a summary document was prepared. The document summarizes both the architecture of ATX and the discussion of proposed changes. You were given this document prior to this experiment to both introduce you to ATX and summarize the meetings.

Tasks

You are joining the development team that is working on evolving ATX. The changes that were discussed during the analysis session are now under development. You will be aiding in that development. Thus, you must understand several key issues in evolving ATX before you can successfully join the team. Please answer the following questions about these issues with as much technical detail as you can. You can use all of the ATX documentation and the video as much or as little as you wish. You should probably spend about 20 minutes researching each question. When looking for something in the video, please think-aloud what you are doing and looking for.

Question 1: What is the difference between the Real Time Database and the Historical Database? What is stored in both and what is that information used for? How do the implementations of the two differ?

Question 2: The designers talked about separating two tasks that currently occur at midnight: the midnight rollover and reconfiguration shutdown. What activities need to occur for each of these two tasks and what are the ramifications of separating them

Question 3: What occurs to the Real Time Database during the current midnight shutdown? How do the midnight rollover and reconfiguration shutdown separately affect the RTDB? What changes will have to occur concerning the RTDB when the midnight rollover and reconfiguration shutdown activities are separated?

SAAMPlayer Post-Task Questionnaire

Thank you for helping me to evaluate SAAMPlayer. Please answer the following questions about your use of SAAMPlayer during your tasks. Please feel free to add any other comments or discuss your comments with me.

1. When completing the tasks, did you rely most heavily on the documentation, the video, or your memory? Why?

2. Did you find anything in the video that you did not already know or find in the documentation? If yes, what?

3. What did you like about using SAAMPlayer?

4. What did you dislike about using SAAMPlayer?

5. What improvements do you think would help make SAAMPlayer easier to use?

Sample TeamSpace Access Behavior Study Tasks

TeamSpace Use Study: Ubicomp group

Name (optional):

Today's Date:

Meeting Date: **August 25, 2004**

Please answer the following questions about last week's meeting. You may use all aspects of MeetingViewer, including the images, to review any information recorded from that meeting. You may also use personal notes, and your memory. You do not have to use TeamSpace if you already know the answer to the question. Please think aloud while you answer the questions – speak the thoughts that you go through as you look for and find the answer. Please answer the questions in order.

1. What was Gregory's thesis topic?

2. Who wants to propose this year?

3. Have we addressed the things we wanted to improve in our group? What do we still need to do?

TeamSpace Post-Task Questionnaire

Thank you for participating in this study. Please answer the following questions about this session with TeamSpace.

1. Name (optional):

2. Today's Date:

3. With 1 being most useful, please rate the following information from most useful to least useful in reviewing this meeting. If you did not find the information useful, do not rate it.

____ Agenda	____ Audio	____ Images
____ Action Items	____ Bookmarks	
____ Presentation/Notes	____ Participants	

4. Is there other information that you would like to review, or use in reviewing, this meeting?

5. How easy or difficult was it to find the information you were looking for?

Very easy   Moderately easy   Not easy or difficult   Moderately difficult   Very difficult

6. What might have facilitated finding information?

7. Do you have any suggestions for improvements to MeetingViewer?

TAGGER Experiment 1 Instructions

In this experiment, you will read a transcript and circle or underline words or phrases that you want to tag. You will use a different pen color for each tag category, and mark each tag with the corresponding letter. The table below is the key or tags and initials.

Tag Key

Terminology tags – blue T – Thing A – Action (Task) C – Condition
Object oriented tags – pink C – Class O – Operation A – Attribute Multiplicity – number ends of links Generalization – G on link
Conversation tags – purple I – Issue IC – Issue, Closed IT – Issue, Tabled R – Resolution RP – Resolution, partial agreement RC – Resolution, consensus agreement RR – Resolution, rejected

The following is an example of how you will tag the transcript:

KA1: Hmm… they want to know when CIM should show masking recommendations. What do you think?

SME1: Well, masking is appropriate if I'm trying to be stealthy and avoid being detected-

KA1: No, I'm pretty sure this is supposed to be in the context of Action on Contact so I think they're assuming that you've already gotten enemy contact.

Instructions

Begin with the terminology tags above. Read the provided transcript, and circle or underline any word or phrase you wish to tag. Next to the word or phrase, write the letter that corresponds to the tag you are asserting. You can connect related tags by drawing a line or arrow between them. Next, do the same for the object oriented tags, and finally for the conversation tags. You will likely have some words or phrases that are tagged multiple times, and thus overlapping lines or circles.

TAGGER Experiment 1 Questionnaire

Date:

Subject #:

Thank you for your participation in this experiment. We are trying to improve the tagging categories and the tagging process. Your feedback is appreciated.

How much software development experience do you have? (Check any that apply)

None: ____ Some commercial (less than 2 years): ______

Some in courses: _______ Moderate commercial (2 to 5 years): ______

Some in research: ____ Experience commercial (more than 5 years): _____

How much software requirements or knowledge acquisition experience do you have? (Check any that apply)

None: ____ Some commercial (one project or experience): ____

Some in courses: ____ Experienced commercial (multiple projects): ____

Which tags did you understand and find reasonable to assign?

Which tags did you not understand or have difficulty assigning? If you can, please explain any confusion.

Do you have any suggestions for changes to the tag set?

Did you find tagging a reasonable process or a difficult one? Please explain your overall impressions of tagging this discussion.

Any other comments:

TagViewer Experiment Instructions

In today's experiment you will be creating a requirements document based on the knowledge acquisition session you previously watched. The purpose of the experiment is to investigate the use of recorded video and in this type of task. You will be given 45 minutes to create a requirements document for an Electronic Checklist that would be integrated with the current flight systems onboard a commercial airplane. You may use your notes, your memory, and the video to complete your requirements. The requirements you identify should be as complete, detailed, and accurate as you can manage. Do not worry about creating a finished document with good formatting, grammar, etc. Instead, focus on getting as many requirements listed, with details that are written concisely but accurately.

Requirements Elicitation Framework

The following is a framework for requirements, along with examples of each taken from the requirements for a desktop meeting scheduler. This framework is to give you an initial structure to work within. However, you are welcome to depart from this framework if you are more comfortable with a different structure, or if you find requirements that do not fit into this structure.

Functional requirement – Describes functions the systems should implement

Example: If the location has not been specified by the person initiating the meeting, the scheduler shall find an appropriate meeting location.

Hardware/Software requirement – Describes how the proposed system needs to interact with existing hardware or software systems.

Example: The scheduler will have interfaces to existing personal digital assistants (PDAs) and computer-aided navigational devices available for automobiles

User interface requirement – Describes an element that the customer wants as part of the user interface, or how the proposed system needs to interact with users.

Example: Meetings that have been marked Urgent should be highlighted in red on the display.

Non-functional requirement – Constrains how the functional requirements are implemented.

Example: If feasible constraints exist for a task, the scheduler shall construct a plan with a time, location, and directions and display a notification to the person initiating the meeting in less than half the time it would take for that person to call one attendee.

Organization Rules / Policies – A rule or policy describes how a work procedure has to happen. These vary from suggested guidelines to mandatory instructions.

Example: All meetings scheduled with senior executives require the approval of that executive or their administrative assistant.

APPENDIX B

OTHER TEAMSPACE USABILITY ISSUES

The evaluation of TeamSpace provided many details on the usability of MeetingViewer. In Chapter 4, we discussed a number of usability issues relating to the use of various indices and navigation behaviors. The following are other usability problems and potential solutions that were encountered during the various evaluations.

- The meaning of the colored event lines on the timeline was sometimes confusing, particularly the participant events. Users wanted these to be speaker identification, instead of merely people joining and leaving the meeting. Also, no one understood what the bookmark events were supposed to be, since we never used them during our meetings. Conveying these meanings will always be a challenge when the reviewer is not familiar with how the capture capabilities and how the events are generated. Additionally, a number of events are summarized in limited space, which limits labeling and explanation. Solutions to this problem could be additional help and some labeling of the timeline events. Labeling the items closest to the current playback point or those within the lower, less dense timeline would be possible and useful.
- There was a mismatch between the user's mental model of notes and the system's implementation. In MeetingClient, a blank presentation was used as a notes page. However, this was labeled as a presentation and located under the Presentation tab in MeetingViewer. Thus, several subjects in the access behavior study forgot about the presence of notes, even after scanning through the whole interface,

because of this labeling. Additionally, on the MeetingViewer presentation panel, only one presentation can be viewed at once. Users did not always realize there were both notes and a presentation, or know how to switch between them. In retrospect, notes should be a separate object and captured and reviewed differently than as an annotation on a slide. We designed MeetingClient with the notion of a whiteboard for brainstorming in mind, not with participants taking notes with a laptop. We did realize that users might want to take personal notes, but purposefully did not implement any private note-taking features.

- Jumping did not start the audio playback. Users often expected audio playback to start as soon as they clicked on any index, such as a slide annotation. Instead, the user is always required to press the play button. Users usually quickly realized this, but even after this realization, some still double, triple, or even quadruple clicked on an annotation or item to move and play.
- The audio was slow to load initially. The audio in TeamSpace does not stream. Instead, it is downloaded in entirety at the first play. This allows for very quick response when jumping and skimming; and users preferred to take the performance hit just once. However, this loading occurred anytime the user pressed stop, and then played again. This reloading is unnecessary. Additionally, audio should download in the background as scanning is occurring, lessening or eliminating any initial delay. Finally, users should also be notified that the audio is downloading and how long that will take. We relied on RealPlayer's notifications, which were not sufficient.

REFERENCES

Abowd, Gregory D. (1999). Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. *IBM Systems Journal,* **38**(4): 508-530.

Abowd, Gregory D., and Elizabeth D. Mynatt (2000). Charting Past, Present, and Future Research in Ubiquitous Computing. *ACM Transactions on Computer-Human Interaction,* **7**(1): 29-58.

Bett, Michael, Ralph Gross, Hua Yu, Xiaojin Zhu, Yue Pan, Jie Yang and Alex Waibel (2000). Multimodal Meeting Tracker. in the *Proceedings of RIAO 2000*, Paris, France, pp.

Bianchi, Michael (1998). AutoAuditorium: a Fully Automatic, Multi-Camera System to Televise Auditorium Presentations. In the *Proceedings of Joint DARPA/NIST Smart Spaces Technology Workshop*, Gaithersburg, MD.

Brotherton, Jason A., Abowd, Gregory D., and Truong, Khai N. (1998). Supporting Capture and Access Interfaces for Informal and Opportunistic Meetings. Technical Report GIT-GVU-99-06, Georgia Institute of Technology.

Brotherton, Jason A. (2001). Enriching Everyday Activities through the Automated Capture and Access of Live Experiences - eClass: Building, Observing and Understanding the Impact of Capture and Access in an Educational Domain. Ph.D. Thesis, College of Computing, Georgia Institute of Technology.

Brotherton, Jason A. and Gregory D. Abowd (2004). Lessons Learned from eClass: Assessing Automated Capture and Access in the Classroom. *ACM Transactions on Computer-Human Interaction (TOCHI),* **11**(2): 121-155.

Brotherton, Jason, J. R. Bhalodia and Gregory D. Abowd (1998). Automated Capture, Integration, and Visualization of Multiple Media Streams. In the *Proceedings of IEEE Multimedia and Computing Systems*, Austin, TX.

Chiu, Patrick, Ashutosh Kapuskar, Sarah Reitmeier, and Lynn Wilcox (1999). NoteLook: Taking Notes in Meetings with Digital Video and Ink. In the *Proceedings of ACM Multimedia '99,* Orlando, FL.

Chiu, Patrick, John Boresky, Andreas Girgensohn and Don Kimber (2001). LiteMinutes: An Internet-Based System for Multimedia Meeting Minutes. In the *Proceedings of Tenth World Wide Web Conference*, Hong Kong, China, pp 140-149.

Conklin, Jeffrey E. (1996). Designing Organizational Memory: Preserving Knowledge Assests in a Knowledge Economy, GDSS.

Conklin, Jeffrey E. and Michael Begeman (1988). "Gibis: A Hypertext Tool for Exploratory Policy Discussion." *ACM Transactions on Office Information Systems,* **6**(4): 303-331.

Cruz, Gil, and Ralph Hill (1994). Caturing and Playing Multimedia Events with STREAMS. In the *Proceedings of ACM Multimedia*.

Degen, L., R. Mander, and G. Salomon (1992). Working with Audio: Integrating Personal Tape Recorders and Desktop Computers. In the *Proceedings of Human Factors in Computing Systems (CHI'92)*, pp 413-418.

Fuchs, Ludwin, Werner Geyer, Heather Richter, Steven Poltrock, Tom Frauenhofer and Shahrokh Daijavad (2001). Enabling Inter-Company Team Collaboration. In the *Proceedings of First International IEEE Workshop on Web-based Collaboration (WetICE'01)*, Boston, MA, pp 374-379.

Geyer, Werner, Heather Richter and Gregory D. Abowd (2003). Making Multimedia Meeting Records More Meaningful. In the *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'03)*, Baltimore, MD, pp 669-672, vol. 2.

Geyer, Werner, Heather Richter, Ludwin Fuchs, Tom Frauenhofer, Shahrokh Daijavad and Steven Poltrock (2001). A Team Collaboration Space Supporting Capture and Access of Virtual Meetings. In the *Proceedings of International Conference on Supporting Group Work (Group'01)*, Boulder, CO, pp 188-196.

Ginsberg, Allen, and Sid Ahuja (1995). Automating Envisionment of Virtual Meeting Room Histories. In the *Proceedings of ACM Multimedia 95*.

Gotel, Orlena C.Z. and Anthony C.W. Finkelstein (1994). An Analysis of the Requirements Traceability Problem. In the *Proceedings of First International Conference on Requirements Engineering (ICRE)*, Colorado Springs, CO, pp 94-101.

Gruber, Thomas R. and Daniel M. Russell (1996). Generative Design Rationale: Beyond the Record and Replay Paradigm. *Design Rationale: Concepts, Techniques, and Use.* T. P. Moran and J. M. Carroll editors, Lawrence Erlbaum Associates, Mahwah, NJ**,** pp 323-349.

Grudin, Jonathan (1994). "Groupware and Social Dynamics: Eight Challenges for Developers." *Communications of the ACM,* **37**(1): 92-105.

Hammond, Tracy, Krzysztof Gajos, Randall Davis and Howard Shrobe (2002). An Agent-Based System for Capturing and Indexing Software Design Meetings. In the *Proceedings of International Workshop On Agents in Design (WAID'02)*, Cambridge, MA.

Haumer, Peter, Klaus Pohl and Klaus Weidenhaupt (1998). "Requirements Elicitation and Validation with Real World Scenes." *IEEE Transactions on Software Engineering,* **24**(12): 1036-1054.

Hayes, Gillian, Shwetak Patel, Khai N. Truong, Giovanni Iochello, Julie Kientz, Rob Farmer and Gregory D. Abowd (2004). The Personal Audio Loop: Designing a Ubiquitous Audio-Based Memory Aid. In the *Proceedings of Mobile HCI*, pp 168-179.

He, Liwei, Jonathan Grudin and Anoop Gupta (2000). Designing Presentations for On-Deman Viewing. In the *Proceedings of Computer Supported Cooperative Work (CSCW'00)*, Philadelphia, PA, pp 127-134.

He, Liwei, Elizabeth Sanocki, Anoop Gupta and Jonathan Grudin (1999). Auto-Summarization of Audio-Video Presentations. In the *Proceedings of ACM Multimedia 1999*, Orlando, FL, pp 489-498.

He, Liwei, Elizabeth Sanocki, Anoop Gupta and Jonathan Grudin (2000). Comparing Presentation Summaries: Slides vs. Reading vs. Listening. In the *Proceedings of Conference on Human-Computer Interaction (CHI'00)*, The Hague, Netherlands, pp 177-184.

Hindus, Debby and Chris Schmandt (1992). Ubiquitous Audio: Capturing Spontaneous Collaboration. In the *Proceedings of Computer Supported Collaborative Work (CSCW'92)*, Toronto, Canada, pp 210-217.

Jaimes, Alejandro, Kengo Omura, Takeshi Magamine and Kazutaka Hirata (2004). Memory Cues for Meeting Video Retrieval. In the *Proceedings of the ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE'04)*, New York, NY, pp 74-85.

Ju, Wendy, Arna Ionescu, Lawrence Neeley and Terry Winograd (2004). Where the Wild Things Work: Capturing Shared Physical Design Workspaces. In the *Proceedings of Computer Supported Cooperative Work (CSCW'04)*, Chicago, IL, pp 533-541.

Kazman, Rick, Reem Al-Halimi, William Hunt and Marilyn Mantei (1996). "Four Paragidms for Indexing Video Conferences." *IEEE Multimedia,* **3**(1): 63-73.

Kazman, Rick, Gregory Abowd, Len Bass, and Paul Clements (1996). Scenario-Based Analysis of Software Architecture. *IEEE Software*.

Kennedy, Sue (1989). Using Video in the BNR Usability Lab. *SIGCHI Bulletin,* **21**(2): 182-185.

Khan, Fawzia (1993). A Survey of Note-Taking Practices. HPL-93-107, Hewlett-Packard Laboratories.

Lee, Dar-Shyang, Berna Erol, Jamey Graham, Jonathan J. Hull and Norihiko Murata (2002). Portable Meeting Recorder. In the *Proceedings of ACM International Conference on Multimedia*, Juan-les-Pins, France, pp 493-502.

Mackay, Wendy E. (1989). EVA: An Experiments Video Annotator for Symbolic Analysis of Video Data. *SIGCHI Bulletin,* **21**(2): 68-71.

Miller, Chris and M Hannen (1999). The Rotorcraft Pilot's Associate: Design and Evaluation of an Intelligent User Interface for a Cockpit Informaiton Manager. *Knowledge Based Systems,* **12**: 443-456.

Minneman, S., S. Harrison, B. Janseen, G. Kurtenback, T. Moran, I. Smith, and B. van Melle (1995). A Confederation of Tools for Capturing and Accessing Collaborative Activity. In the *Proceedings of ACM Conference on Multimedia (Multimedia'95)*.

Moran, Thomas P., Leysia Palen, Steve Harrison, Patrick Chiu, Don Kimber, Scott Minneman, William van Melle, and Polle Zelweger (1997). "I'll Get That Off the Audio": A Case Study of Salvaging Multimedia Meeting Records. in the *Proceedings of Human Factors in Computing Systems (CHI'97)*, pp 202-209.

Moran, Thomas P., Patrick Chiu, and William van Melle (1997). Pen-Based Interaction Techniques for Organizing Material on an Electronic Whiteboard. In the *Proceedings of User Interface Software and Technology (UIST '97)*.

Mukhopadhyay, S. and B. Smith (1999). Passive Capture and Structuring of Lectures. In the *Proceedings of ACM Multimedia '99,* Orlando, FL.

Muller, R., and T. Ottmann (1998). The "Authoring on the Fly" System for Automated Recording and Reply of Telepresentations. *ACM/Springer Multimedia Systems Journal, special issue on Multimedia Authoring and Presentation Techniques,* **8**(3).

Plaisant, Catherine, Brett Milash, Anne Rose, Seth Widoff and Ben Shneiderman (1996). LifeLines: Visualizing Personal Histories. In the *Proceedings of Human factors in computing systems (CHI '96)*, Vancouver, British Columbia, Canada.

Richter, Heather and Gregory D. Abowd (1999). Automating the Capture of Design Knowledge: a Preliminary Study. Technical Report GVU-99-45, Georgia Institute of Technology.

Richter, Heather, Gregory D. Abowd, Werner Geyer, Ludwin Fuchs, Shahrokh Daijavad and Steven Poltrock (2001). Integrating Meeting Capture within a Collaborative Team Environment. iIn the *Proceedings of UbiComp 2001*, Atlanta, GA.

Richter, Heather, Jason Brotherton, Khai N. Truong and Gregory D. Abowd (1999). Multi-scale Timeline Slider. Technical Report GVU-99-30, Georgia Institute of Technology.

Richter, Heather, Chris Miller, Gregory D. Abowd and Harry Funk (2004). Tagging Knowledge Acquisition to Facilitate Knowledge Traceability. *International Journal of Software Engineering and Knowledge Engineering,* **14**(1): 3-19.

Richter, Heather, Pascal Schuchhard and Gregory D. Abowd (1999). Automated Capture and Retrieval of Architectural Rationale. Position paper at the *First IFIP Working Conference on Software Architecture*, San Antonio, TX.

Roschelle, J., R. Pea and Randall Trigg (1990). VideoNoter: a Tool for Exploratory Video Analysis. IRL90-0021, IRL.

Schultz, Tanja, Alex Waibel, Michael Bett, Florian Metze, Yue Pan, Klaus Ries, Thomas Schaaf, Hagen Soltau, Martin Westphal, Hua Yu and Klaus Zechner (2001). The ISL Meeting Room System. In the *Proceedings of Workshop on Hands-Free Speech Communication (HSC-2001)*, Kyoto Japan.

Selvin, Albert, Simon Buckingham Shum, Maarten Sierhuis, Jeffrey E. Conklin, Beatrix Zimmermann, Charles Palus, Wilfred Drath, David Horth, John Domingue, Enrico Motta and Gangmin Li (2001). Compendium: Making Meetings into Knowledge Events. In the *Proceedings of Knowledge Technologies*, Austin, TX.

Stifelman, L.J. (1992). Augmenting real-world objects: A paper-based audio notebook. In the *Proceedings of Human Factors in Computing Systems (CHI'92)*, pp 199-200.

Stifelman, Lisa, Barry Arons, and Chris Schmandt (2001). The Audio Notebook: Paper and Pen Interaction with Structured Speech. In the *Proceedings of ACM Computer-Human Interaction (CHI 2001)*, Seattle, WA, pp 182-189.

Truong, Khai N., Gregory D. Abowd and Jason Brotherton (2001). Who, What, When, Where, How: Design Issues of Capture and Access Applications. In the *Proceedings of Ubicomp 2001*, Atlanta, GA.

Uchihashi, Shingo, Jonathan Foote, Andreas Girgensohn and John Boreczky (1999). Video Manga: Generating Semantically Meaningful Video Summaries. In the *Proceedings of International Conference on Multimedia*, Olrando, FL, pp 383-392.

Weber, Karon and Alex Poon (1994). Marquee: A Tool For Real-Time Video Logging. In the *Proceedings of Human Factors in Computing Systems (CHI'94)*, Boston, MA.

Whittaker, Steve, Patrick Hyland, and Myrtle Wiley (1994). Filochat: Handwritten notes provide access to recorded conversations. In the *Proceedings of Human Factors in Computing Systems (CHI'94)*, pp 271-277.

Wilcox, Lynn D., Bill N. Schilit, Nitin Sawhney (1997). Dynomite: A Dynamically Organized Ink and Audio Notebook. In the *Proceedings of Conference on Human-Computer Interaction (CHI'97)*, Atlanta, GA.

Wolf, Catherine G., and James R. Rhyme (1992). Facilitating Review of Meeting Information Using Temporal Histories. RC 19811, IBM Research.

Wolf, Catherine G., James R. Rhyne, Laura K. Briggs (1992). Communication and Information Retrieval with a Pen-based Meeting Support Tool. In the *Proceedings of Computer Supported Collaborative Work (CSCW'92)*.

Wood, D.P., M.G. Christel and S.M. Stevens (1994). A Multimedia Approach to Requirements Capture and Modeling. in the *Proceedings of First International Conference on Requirements Engineering (ICRE)*, Colorado Springs, CO, pp 53-56.