Automating the capture of design knowledge: a preliminary study

Heather A. Richter & Gregory D. Abowd
College of Computing
801 Atlantic Dr.
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
+1 404 385 1101
{hrichter, abowd}@cc.gatech.edu

ABSTRACT
A large amount of design information that is generated during design often does not get recorded, resulting in a potential loss of important design knowledge. The design rationale community has offered methods for capturing and documenting this important design knowledge, but these methods have not succeeded in practice. We propose using ubiquitous computing technology to automatically capture design information as it is generated naturally in design meetings. In this paper, we report the results of a preliminary case study examining how automated capture can be used to produce multimedia records of design discussions.

Keywords
Design rationale, ubiquitous computing, automated capture, design knowledge, SAAM

1 INTRODUCTION

A large amount of design information that is generated during design often does not get recorded in formal design documentation. Some of this information is often referred to as design rationale, but can include any sort of knowledge of the who, what, when, where, why, and how of design. The design rationale community has long attempted to create methods and notations for capturing some of this extensive design information. However, these attempts have not become common practice. In section 2, we summarize this previous work and argue that these failures are twofold: first, the traditional rationale methods and notations are too time-consuming for designers, and second, these methods are not sufficiently rich to capture the full complexity of design rationale. What developers want instead are methods to capture design knowledge cheaply and as completely as possible.

Software design and evolution practices consist of many discussions among groups of designers. These discussions are often rich in design information: decisions are made, alternatives are considered, and details are explained. Ubiquitous computing has as a major theme the automated capture and access of such discussions. We propose applying this technology to design discussions, thereby capturing the rich, informal design knowledge that is generated from natural design activities.

Capturing design meetings has two potential uses. In the short term, automated capture can give designers a better memory of their meetings. In the long term, the capture can help maintainers or those responsible for evolving the software to gain a better understanding of the system by having access to richer set of design information. In order to understand how to capture the efforts of software designers, we must first understand how real developers with real tasks can utilize the captured information. Section 3 describes a preliminary case study we are using to examine the potential of design meeting capture. In section 4, we describe a small experiment we performed with this case study to gain a better understanding of the issues involved in effectively using captured design information. The results and analysis of this experiment are presented in sections 5 and 6. In section 7, we discuss several of the general issues we discovered in making ubiquitous capture of design knowledge usable and useful. Finally, Section 8 concludes and summarizes our plans for further investigation of this approach.

2 BACKGROUND

Design Rationale

Design rationale is the explanation behind the design - why the design is the way it is. Rationale can include assumptions made about the system, the alternatives considered, and the reasoning behind decisions. Research in design rationale has primarily focused on languages for describing rationale and associated methods for capturing the information. Traditional approaches to rationale capture have involved two methods for documenting and

Submitted for review to the International Conference on Software Engineering (ICSE'00)

structuring design rationale: process-oriented and structure-oriented. A process-oriented approach, such as Issue Based Information Systems (IBIS) (Kunz 1970; Conklin 1996), focuses on documenting rationale as it occurs during design meetings. A structure-oriented approach, such as Questions, Options, and Criteria (QOC) (MacLean 1996), focuses instead on a post hoc structuring of the rationale to show the complete design argument. Process-oriented approaches potentially interrupt the design process by requiring additional work for the designers that is not natural. Designers must take detailed notes of their discussions in the terms dictated by the notation. Thus, process-oriented rationale capture techniques need to make capture as transparent as possible. While structure-oriented approaches won't interrupt design work, important rationale may be lost in summarizing it after the fact. Both approaches require designers to continuously maintain the rationale as the design changes. Thus, the amount of work required to document design rationale must be justified in order for a particular technique to be successful.

In the early 1990s, design rationale was a prominent research topic in Human-Computer Interaction. Projects attempted to improve rationale languages and create tools for capturing and using the rationale. These projects highlighted several challenges in design rationale capture. The first challenge is the difficulty of organizing and viewing the large volume of information. A second issue is that the design rationale needs to be linked to the concrete artifacts that are used and produced in the design. Despite these research efforts, the proposed methods of design rationale capture, even with tool support, are time-consuming and difficult to maintain. This cost is preventing acceptance of design rationale in software development.

As Gruber and Russel state (Gruber 1991), the task of eliciting, recording and organizing design knowledge is difficult. Creating and collecting rationale is even more difficult. Rationales are explanations of the relationships between many different aspects of the design, such as the structure, behavior, artifacts and the decision-making process. Gruber and Russel concluded that users of rationale did not merely look up rationale information, but instead constructed or inferred it from a variety of information sources. Thus, rationale capture shouldn't focus on forming complete answers, but instead on recording all of the data that could be used to reconstruct those answers later. However, design rationale techniques have focused on capturing rationale as complete explanations using semi-formal structures. In doing so, they fail to capture the full richness and complexity of rationales and will never be sufficient to answer all rationale-related questions.

Ubiquitous Computing

One of the potential features of a ubiquitous computing environment is that it could be used to record our everyday experiences and make that record available for later use. Indeed, we spend much time listening to and recording, more or less accurately, the events that surround us, only to have that one important piece of information elude us when we most need it. We can view many of the interactive experiences of our lives as generators of rich multimedia content. A general challenge in ubiquitous computing is to provide automated tools to support the capture, integration and access of this multimedia record. The purpose of this automated support is to have computers do what they do best, record an event, in order 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. If we consider the many software design meetings and discussions as these live experiences, then some of the information that would be recorded is design knowledge that comes out of the natural activities of designers.

Ubiquitous capture environments have been built for several domains: education, personal note-taking, and meeting capture. The Classroom 2000 project is mainly concerned with capture, integration and access in support of lecture based education (Abowd 1999a). The many streams of activity in a typical lecture — what is spoken, what is seen, what is written down on a whiteboard and what is shown on public displays — are combined to provide a rich interactive experience that is becoming increasingly more difficult to capture using traditional pen and paper notes. Work on the Marquee system at PARC (Weber 1994) together with work on the Filochat system at Hewlett-Packard (Whittaker 1994), Apple (Degen 1992), and MIT's Media Lab (Stifelman 1992) demonstrates the utility of personal note-taking with automatic audio enhancement for later review. Research in ubiquitous meeting capture has mostly focused on promoting collaboration through shared work surfaces. The Dolphin (Streitz 1994) and Tivoli (Minneman 1995; Moran 1997) systems provide shared surfaces with various tools for drawing and manipulating content that can later be reviewed with audio enhancement.

Ubiquitous capture of design knowledge has several benefits over traditional rationale capture techniques. First, it is cheap in terms of the effort of designers. In fact, the capture should be invisible to the designers, requiring no change in their activities. Second, it captures the full richness of design information as it is generated by not imposing a predefined structure on that information. Instead, the technology can use the natural structure of software design meetings to help provide meaningful ways for designers to access the captured information.

3 CASE STUDY

To examine ubiquitous capture of design meetings, we first focused on capturing a particular kind of 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.

As a preliminary case study, we facilitated and video recorded a SAAM analysis with a group of designers at Nortel. 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 has 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. The authors facilitated and participated in the meetings, while another graduate student ran the video camera. The six Nortel 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 Nortel.

Next, we created a prototype interface for browsing and viewing the digitized video of the meetings. This interface, called SAAMPlayer, is shown in Figure 1. SAAMPlayer consists of a video window, simple playback controls, and a timeline for video browsing. SAAMPlayer uses the RealVideo™ plug-in to provide streaming video playback. The black diamond scrub on the timeline shows where playback is occurring. The user can move this scrub to easily replay any segment of the meetings.

We decided that an electronic version of the summary document could serve as a natural index into the video. We manually created 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. SAAMPlayer uses different colors to distinguish different keywords on the timeline. For example, in Figure 1, the keyword "RTDB" is red, "HDB" is yellow, "midnight" is blue, and "system reconfiguration" is green. The lines drawn above and below the timeline show all the points where these keywords were mentioned. Browsing the video is then a process of using keywords to find relevant sections of the video to replay.

The first problem we discovered in creating the prototype was that of scale. A traditional single-scale timeline, as shown in Figure 1, was not sufficient for browsing even this set of meetings, which was 5 ½ hours long. The user did not have precise enough control over where to start playback of the video. Additionally, groups of time-points for the keywords can be indistinguishable at such a scale. Thus, we created a new timeline widget, the Multi-Scale Timeline Slider (MTS) to provide focus + context browsing (Richter 1999a). The use of MTS is shown in Figure 2, where the top timeline is the same as the timeline shown in Figure 1. Each subsequent timeline is a focused region of the previous timeline. The focus region is represented as the area between two sliding bars. Each bar can be dragged to widen or narrow the region of focus. Additionally, the user can drag the whole region using the top bar of the focus region. Note that information is not added as the timeline becomes more focused. However, as focus increases, the individual annotations spread out and are easier to distinguish. On the more focused timeline, the user also has finer control over where to replay the video.

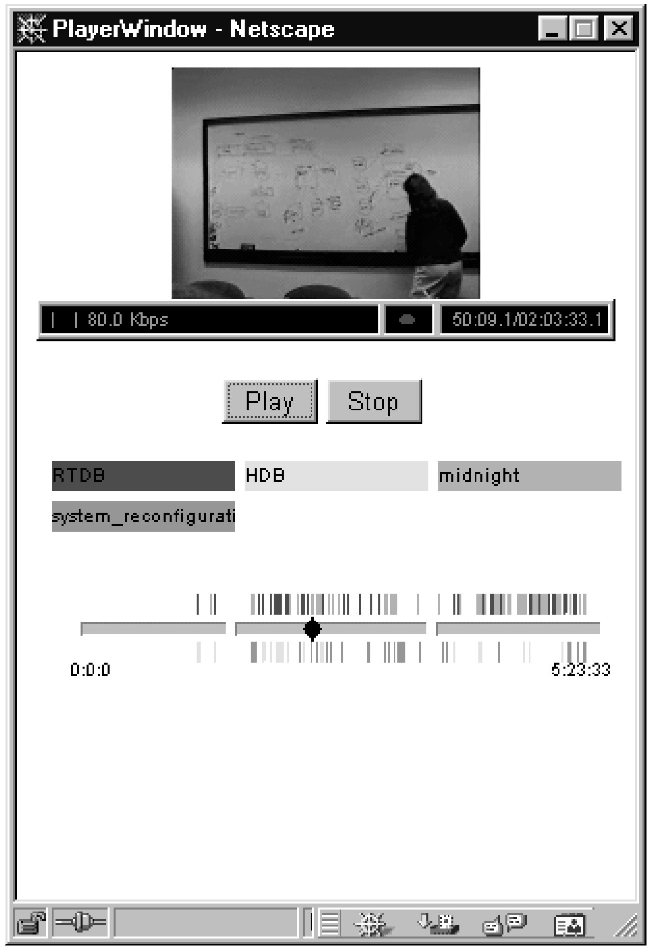

Figure 1. SAAMPlayer prototype. (A screenshot of the summary document is not included here due to the sensitivity of its contents, but will be shown in the final version of this paper.)

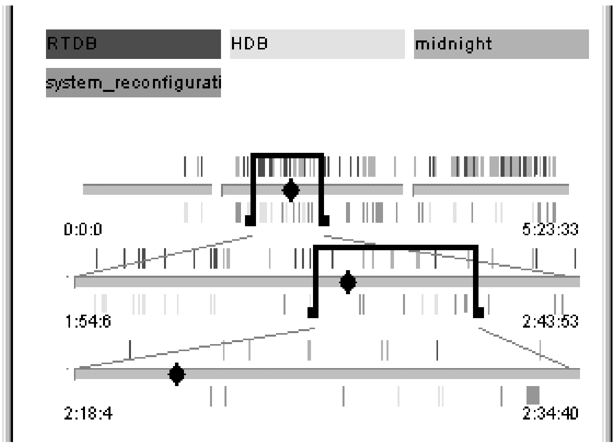

Figure 2. Multi-Scale Timeline Slider in use on SAAMPlayer.

4 EXPERIMENT

We ran a small experiment to examine the use of SAAMPlayer and determine its effectiveness for accessing design meetings. We wanted to observe the facility with which users could find information in the video. There were two sets of subjects for this trial: subjects who were in the original SAAM meetings, and subjects who had no prior knowledge of the Nortel system. Subjects 1 and 2 were the Georgia Tech professor and one of the graduate students who participated in the original meetings. The memories of these two were foggy at best, as the experiment took place nine months later and neither had looked at the summary document or video since the original meetings. Subjects 3 and 4 were software engineering graduate students. All of the graduate students have several years of experience as software developers in industry.

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 Nortel system 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 the video. We asked the subjects to think aloud when they were browsing the video. Following the task, users were asked for feedback on using SAAMPlayer.

5 RESULTS

All subjects used both the document and the video to answer the questions. In the feedback, they expressed that they used the document as a starting point for answering questions and then looked to the video for additional information or clarification. Subjects 1, 2, and 3 stated that they did find information in the video that they did not find in the document. For example, subject 3 found information on how internal applications use a particular database of information while external applications use a replicated archive. Subject 2 found information on a particular system memory constraint which was key to his answers. However, subject 4 did not find new information. As she stated, the video “reinforced what was already documented.” Subject 3 also stated that he liked to hear what was “actually said” beyond just relying of the document for information.

As stated earlier, browsing the video consists of highlighting keywords and playing back relevant sections. However, each user did this slightly differently. Figure 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

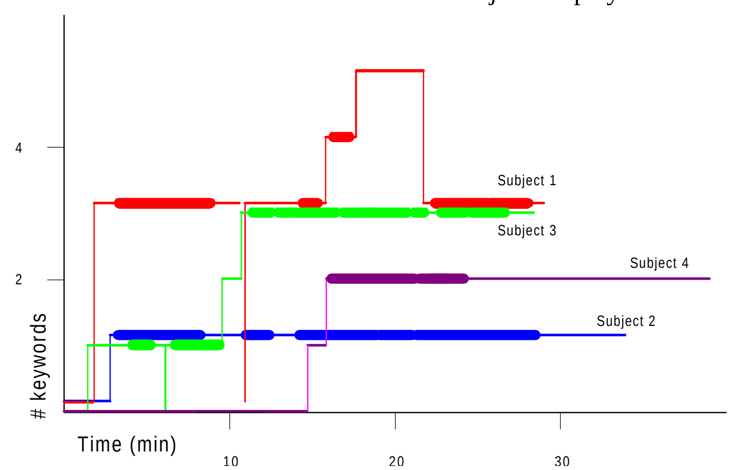

Figure 3. Browsing behavior of number of keywords vs. time. Lines are wider where subjects are playing video.

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 were 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.

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.

6 ANALYSIS

Video utility

In this 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.

Browsing strategies

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. In their feedback, all subjects stated that they liked this feature of SAAMPlayer. The second strategy was quickly skimming areas that looked relevant. Subjects would start early in the meeting and skip through a set of relevant points. Additionally, subjects who attended the meetings used their memory of what occurred to help them determine when information might be useful. For example, they remembered that the first meeting contained very general information, but that the second meeting contained more detailed analysis of the architecture, and that the third meeting contained details of the impact of certain changes. Subject 3, who did not attend the meeting, also employed these strategies by using the document to figure out the possible order of activities.

This study also points to strategies that could be employed with improvements to SAAMPlayer. Browsing might be improved by providing links to the artifacts. In one discussion, subject 1 heard a reference to "those components in the top left-hand corner" but was unable to see these clearly enough on the video to know which components those were. Subjects also expressed a desire to more tightly couple portions of the document with the video. Subject 3 wanted to be able to click on sentences and see what discussion that sentence summarized. He also wanted to search for additional keywords that were not provided.

Context

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, which is often referred to as context. Some of these activities are meeting specific, such as whether the discussion was determining the architecture or evaluating the impact of a particular change. Subjects 1, 2, and 3 guessed where these activities might be occurring, but expressed desire to actually see these boundaries. Other activities are more general. Subject 4 wanted to know where keywords were defined. Subject 1 wanted to see where an issue was resolved. Other context could be user-generated. Subject 1 wanted to mark points he found particularly interesting. If SAAMPlayer could have highlighted already played portions of video, subject 3 wouldn't have accidentally repeated segments.

Scale

Even in this small study, scale proved to be important. The three subjects who were able to focus the timeline had no difficulty controlling their playback of the video. However, subject 4, who did not use focus, wasted time trying to find the part of the discussion that contained the keyword. On the top-level timeline, each pixel represents over one minute in the video, whereas on a lower timeline each pixel might represent 10 seconds. Thus, when subject 4 tried to position the playback at a particular keyword, she was often actually starting the video minutes away.

7 UBIQUITOUS COMPUTING ISSUES

The challenge in applying ubiquitous computing to capturing design knowledge is to understand how the captured information can be used later by developers and then how that information can automatically be captured and presented in the appropriate form. This requires going beyond simple record and playback schemes to providing meaning to the various activities and records of a design meeting. We believe our initial case study, while small and simple, illustrates the potential use of ubiquitous capture of design information. However, this novel technology needs to be put to real use to fully understand the implications. Our case study, as well as previous design rationale capture research, point out several larger issues we need to address in order to be successful.

The first issue, scaling to a large volume of information, was an issue in our simple prototype and was also mentioned as a failing of traditional design rationale techniques. Browsing and retrieving information in just a few hours of video required novel visualizations. If many meetings were recorded, there is the potential for hundreds of hours of audio, video, and other recorded activities. First, this much information must be viewable in a reasonable amount of screen space. Second, more powerful browsing mechanisms need to be provided to search over all this information.

The ability to navigate successfully such a large volume of information leads to the need for recording and visualizing additional context. The more information the user has about what was occurring during the meeting, the more they will understand where to find the information they need. Examples of context can include who is speaking, what is said, or what activity is happening. These activities might be general to any meeting, such as brainstorming, drawing a diagram, or following an agenda. Additionally, the context could be meeting-specific. In our case study, there were a certain set of activities that occurred, with different design information being discussed in each. Additionally, users may wish to provide their own context. Users might wish to annotate the video to highlight portions they have already visited or portions they find particularly interesting. Different context might also be important for different kinds of retrieval tasks. We first need to explore what contexts are useful in helping users navigate and find information. We then need to explore how these contexts can be captured automatically during the design meeting or added later by designers.

Software designers have a very specific goal in mind: designing a system. The designers are often working with and producing standard artifacts. Any knowledge capture should help them capture these artifacts, as well as link other captured information to other relevant artifacts. Thus, ubiquitous capture will not just involve video-taping meetings, but perhaps recording drawings, gestures, web site visits, and numerous other artifacts. For example, in our case study, the main artifact we produced is a software architecture, which we drew as a box and line diagram. If the system understood this, it would better understand the manipulation of that diagram and how to link the diagram to the other activities. In previous work we provided a system that would help users draw and manipulate such an architectural drawing and then link the recorded discussion to those activities (Richter 1999b). Additionally, designers will be producing many different kinds of documentation to later serve as information sources. Thus, the browsing of hours of recorded video will be occurring within a higher level task of browsing many different kinds of related multimedia documents.

Finally, we must address the automation of this whole capture process. The capture and integration of information should essentially be invisible to the designers. The more the system understands about the activities of the users, the more context it can provide later to help users find information. For this study, the recording, digitizing, and time-stamping of the video were manual. However, the technology does currently exist to perform these activities automatically, as demonstrated in the Classroom 2000 system. Ongoing research in keyword spotting, transcribing, gesture recognition and handwriting recognition could be utilized. In addition, the ability to extract context — the who, what, where, when and why of a situation — is an active research topic in ubiquitous computing (Abowd 1999b) and has clear implications in facilitating the tagging of captured information with relevant meta-information that supports later retrieval tasks.

8 CONCLUSIONS AND FUTURE WORK

We believe that ubiquitous capture technology can cheaply and effectively be used to record a rich, informal record of design activities. This record can be used to provide additional documentation to help later designers better understand the system and the effects of their own activities. We reported the results of a preliminary case study to examine this approach. In our experiment, users were able to successfully browse and find information in video recorded meetings. The study also identified issues of scale, browsing strategies, and context as challenges to successfully utilizing captured design meetings. We can not fully explore these issues until we apply this technology to realistic use and evaluate the effect on designers' everyday activities.

We wish to take our future work in two directions. First, we need a large, realistic study where we can record many meetings, and where real users will later have real needs of understanding the system. In the preliminary case study, the evaluators seemed to think having the video was valuable. But would they use it in real work tasks and how can the information best help them? What information is important to capture in these situations? In order to examine these questions, we need to have many more hours of recorded activities than our first study.

Secondly, we also wish to focus on a particular type of meeting to investigate the kinds of context we can capture and understand based on the particular structure and artifacts of the meeting. In our preliminary study we focused on an architectural analysis. However, we may need to start with meetings that are less abstract with more consistent artifacts. We are currently examining future case studies for both of these directions.

9 REFERENCES

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

Abowd, G. D. (1999b). Software Engineering Issues for Ubiquitous Computing. International Conference on Software Engineering (ICSE'99), IEEE Computer Society Press.

Conklin, E. J., and KC Burgess-Yakemovic (1996). A Process-Oriented Approach to Design Rationale. Design Rationale: Concepts, Techniques, and Use. T. P. Moran, and John M. Carroll. Mahwah, NJ, Lawrence Erlbaum Associatesq**:** 393-427.

Degen, L., R. Mander, and G. Salomon (1992). Working with Audio: Integrating Personal Tape Recorders and Desktop Computers. Human Factors in Computing Systems (CHI'92).

Gruber, T. R. and. D. M. R. (1991). Design Knowledge and Design Rationale: A Framework for Representation, Capture, and Use, Knowledge Systems Laboratory.

Kazman, R., G. D. Abowd, L. Bass, and P. Clements (1996). "Scenario-Based Analysis of Software Architecture." IEEE Software **13**(6): 47-56.

Kunz, W. M., and H. Rittel (1970). Issues as elements of information systems. Berkeley, University of California, Berkeley, Institute of Urban and Regional Development.

MacLean, A., R. M. Young, V. M.E. Bellotti, and T. P. Moran (1996). Questions, Options, and Criteria: Elements of Design Space Analysis. Design Rationale: Concepts, Techniques, and Use. T. P. Moran, and John M. Carroll. Mahwah, NJ, Lawrence Erlbaum Associates**:** 53-105.

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. ACM Conference on Multimedia (Multimedia'95).

Moran, T. P., L. Palen, S. Harrison, P. Chiu, D. Kimber, S. Minneman, W. van Melle, and P. Zelweger (1997). "I'll Get That Off the Audio": A Case Study of Salvaging Multimedia Meeting Records. Human Factors in Computing Systems (CHI'97).

Richter, H. A., J. A. Brotherton, G. D. Abowd, and Khai N. Truong (1999a). A Multi-Scale Timeline Slider for Stream Visualization and Control, Georgia Institute of Technology.

Richter, H. A., P. Schuchhard, and G. D. Abowd (1999b). Automated Capture and Retrieval of Architectural Rationale. Position paper in First IFIP Working Conference on Software Architecture

Stifelman, L. J. (1992). Augmenting real-world objects: A paper-based audio notebook. Human Factors in Computing Systems (CHI'92).

Streitz, N. A., J. Geibler, J.M. Haake, and J. Hol (1994). DOLPHIN: Integrated Meeting Support across Local and Remote Desktop Environments and LiveBoards. Computer Support Collaborative Work (CSCW'94).

Weber, K., and A. Poon (1994). Marquee: A tool for real-time video logging. Human Factors in Computing Systems (CHI'94).

Whittaker, S., P. Hyland, and M. Wiley (1994). Filochat: Handwritten notes provide access to recorded conversations. Human Factors in Computing Systems (CHI'94).

Wolf, C.G., J.R. Rhyne, L.K. Briggs (1992). Communication and Information Retrieval with a Pen-based Meeting Support Tool. Computer Supported Collaborative Work (CSCW'92).