Who, What, When, Where, How: Design Issues of Capture & Access Applications

Khai N. Truong, Gregory D. Abowd, and Jason A. Brotherton

College of Computing & GVU Center, Georgia Institute of Technology
Atlanta, Georgia 30332-0280, USA
{khai,abowd,brothert}@cc.gatech.edu

Abstract. One of the general themes in ubiquitous computing is the construction of devices and applications to support the automated capture of live experiences and the future access of those records. Over the past five years, our research group has developed many different capture and access applications. In this paper, we present an overview of six of these applications. We discuss the different design issues encountered while creating each of these applications and share our approaches to solving these issues (in comparison and in contrast with other work found in the literature). From these issues we define the large design space for automated capture and access. This design space may then serve as a point of reference for designers to extract the requirements for systems to be developed in the future.

1 Introduction

In his seminal 1991 Scientific American article, Mark Weiser describes a vision of ubiquitous computing in which technology is seamlessly integrated into the environment and provides useful services to humans in their everyday activities [34]. Over the years, one of the services envisioned is the automated capture of everyday experiences made available for future access. Automated capture and access applications leverage what computers do best – record information. In return, humans are free to fully engage in the activity and to synthesize the experience, without having to worry about tediously exerting effort to preserve specific details for later perusal.

This research theme is not unique to ubiquitous computing. Vannevar Bush was perhaps the first to write about the benefits of a generalized capture and access systems when he introduced the concept of the memex [8]. The memex was intended to store the artifacts that we come in contact with in our everyday lives and the associations that we create between them. Over the years, many researchers have worked towards this vision. As a result, many systems have been built to capture and access experiences in classrooms [2, 3, 20], meetings [4, 9, 10, 12, 13, 19], and other generalized experiences [23, 29, 32]. In our research group, we have also looked at how the capture and access of experiences can assist people in a variety of situations

G. D. Abowd, B. Brumitt, S. A. N. Shafer (Eds.): Ubicomp 2001, LNCS 2201, pp. 209-224, 2001.

including college lectures [1, 31], software engineering design meetings [26], impromptu meetings [6], military strategic planning sessions, academic conferences [11], distributed meetings [25], and inside the home.

We define *capture and access* as the task of preserving a record of some live experience that is then reviewed at some point in the future. Capture occurs when a tool generates an artifact that documents the history of what happened. The artifacts, or *captured data*, are recorded as *streams of information* that flow through time [7]. The tools that record experiences are the *capture devices*; and the tools used to review captured experiences are the *access devices*. A *capture and access application* can exist in the simplest form through a single capture and access device or in a more complex form as a collection of capture and access devices [18]. Under our definition, some tools are already inherent capture and access devices; e.g. pen and paper, cameras and camcorders. However, some of these tools only support a single user during the capture of the information. Others limit access to occur at only a single location at a time. More compelling applications are often built to support a larger community with more universal access.

In this paper, we present a design space for capture and access applications. We map out five dimensions in this design space and discuss the key attributes of each dimension. Six different case studies (in the form of overviews of work we have created in the past) are used to formulate this design space. As we describe each application, we compare and contrast our approach for building each application with other work found in the literature. In doing so, we show that these issues we present are ones other researchers have faced too. By discussing multiple approaches we are able to discuss tradeoffs between these approaches. Collectively, this design space and these six case studies can act as a point of reference for future capture and access application designs.

2 Review of Capture and Access Applications

Over the past five years, we have explored the use of capture and access in a variety of situations. Different applications have been built for different environments. In this section we will overview and discuss the approaches taken to build these applications. We begin with the human-centered motivation for each capture and access application, then describe briefly some details of the system we have built and relate our application to others reported in the literature. All of these applications motivate our desire to provide a more general description of the design space of capture and access applications.

2.1 Classroom 2000

2.1.1 Motivation

As technology has been introduced into classrooms, instructors are given the ability to present more information during each lecture, with the goal of providing a richer learning experience. As a result, students are often drowned with information and forced into a “heads down” approach to learning. While students are busy copying down everything presented in class, they are potentially distracted from paying attention to the lecture itself. An instructor produces a lot of artifacts while teaching (lecture slides, handwritten annotations, and spoken words), which students attempt to preserve in their notes. The Classroom 2000 project aimed to alleviate some of the student’s burden by recording much of the public lecture experience [1, 7].

2.1.2 How It Works

To capture what the instructor writes, we used electronic whiteboards (e.g., the LiveBoard [12] or a SmartBoard [27]). For instructors who teach with a prepared presentation, we converted the presentation into slides displayed on the electronic whiteboard that can be written on; otherwise, it acts as a simple whiteboard. To capture what the instructor says and does, the classroom contains microphones used to record the audio and a single camera to capture a fixed view of the classroom. Finally, to capture other web accessible media the instructor may want to present, a web proxy was used to monitor and record the web pages visited during each class.

Immediately after each class, all the different captured streams of information are processed to create an on-line multimedia-augmented set of lecture notes in a form that supports student review. In order to build the appropriate access interface, we considered when and where most studying would occur. While classes are regularly scheduled activities that occur in specific rooms (thereby, specifying when and where capture occurs), when and where students review the notes will largely vary. Because the notes are multimedia enhanced, we require an electronic format. We decided that it would make sense that the notes are available in a web-accessible format. This infrastructure would also allow students to review the notes at their own convenience.

2.1.3 Related Work

Rather than instrumenting the classroom with augmented capture devices (such as the LiveBoard as an augmented whiteboard and PCs that pull web pages from a logging web proxy), the Lecture Browser application [20] and other whiteboard applications such as the ZombieBoard [5] and BrightBoard [28] rely on cameras and vision techniques to capture the materials written and presented on the boards, as well as to detect changes. The tradeoff between these two approaches lies in the granularity of capture as well as the level of intelligence built in to the capture systems. By instrumenting the physical objects the user interacts with, we are able to obtain a finer level of granularity in the interaction history without needing to apply much intelligence into the system. For example, when the instructor writes on the electronic whiteboard, we can easily access information at the stroke level. Capture devices that rely on machine vision face a greater challenge to extract this level of information. For example, occlusion by the lecturer can prevent the system from seeing all of the writing as it is being written. As a result, the change detected is not a stroke level, but at a cluster level (or a coarser level of granularity).

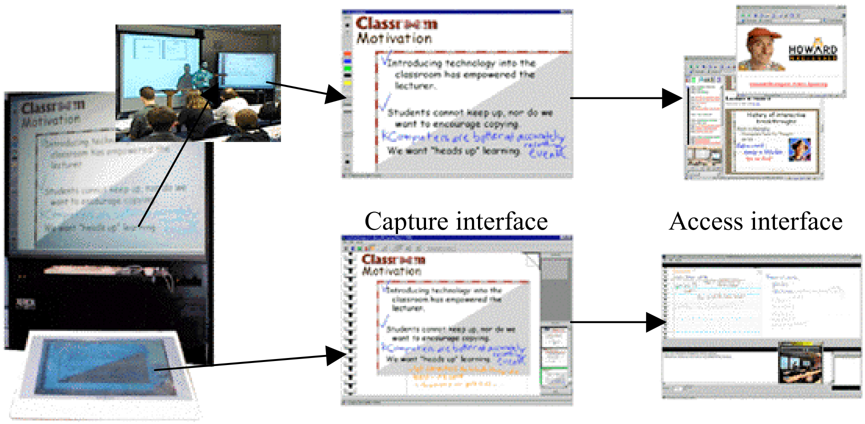

Fig 1. Classroom 2000 & StuPad. At the top of the figure is shown the public capture of Classroom 2000. The bottom portrays the personalized augmentation of the public capture provided by StuPad

2.2 StuPad (Student NotePad)

2.2.1 Motivation

The goal of Classroom 2000 was to help relieve students from needing to tediously copy down all the notes presented during class. However, because Classroom 2000 lecture notes are captured through actions performed strictly by the instructor(s), it excludes students from being able to make the notes personally meaningful. As a result, some students still take a small amount of private notes with pen and paper. When students study, they are forced to manually integrate the electronically automatically captured lecture notes (provided by Classroom 2000) with their own private notes on paper – a nontrivial task. To better support the integration of each student's notes with the Classroom 2000 notes, student notes needed to also be in an electronic format that could be synchronized with the other captured streams of information. We wanted to create a system that integrates the public streams of information captured by Classroom 2000 into each student's electronic notebook during the lecture, giving students the ability to personalize the material as it being captured.

2.2.2 How It Works

To support the personalization of the captured experience, we built the Student Notepad (or StuPad) system to provide students with an interface that is capable of integrating the prepared presentation, digital ink annotations and Web pages browsed from the public classroom notes into each student's private notebook (during the capture phase). Relatively affordable video tablet technology allowed us to instrument the students' desks with a note-taking environment that is networked and at least as powerful as the electronic whiteboard at the front of the class.

During class, the act of writing is more natural for students to perform and less distracting than typing. Outside of class, it is hard to predict when and where students will review the notes; therefore, the access application was designed to run on networked computers with the more traditional keyboard/mouse interface. The personalized notes are reviewed over the web to facilitate students to be able to review the notes anywhere anytime.

2.2.3 Related Work

Other systems, such as the Audio Notebook [29] and Dynomite [36] are private systems used only to capture an experience for just that individual. NotePals [10] allows users to each privately capture their notes during the live experience. After the experience, all the users notes are gathered to form a collective view of the experience during the access phase. This approach takes into consideration that some points may be missing in different people's notes, or that the users' views may be different. The NoteLook system [9] also supports the integration of both public and private content. The NoteLook system provides users with an array of camera views that can be used to take snapshots of the public presentation when a seminar participant requests. Once the snapshot is integrated into the user's private notebook, private annotations can be placed on top of it. The subtle difference between NoteLook and StuPad lies in NoteLook's reliance on the participants to devote effort and awareness (as well as a little anticipation) on when to request the public information to be added into their personal notebook.

2.3 SAAMPad (<u>S</u>oftware <u>A</u>rchitecture <u>A</u>nalysis <u>M</u>ethod <u>Pad</u>)

2.3.1 Motivation

Software evolution is a difficult and time-consuming software development activity that can be enhanced when designers have a more complete understanding of the rationale underlying the current architecture for an existing system and the implication of any changes resulting from the evolution. To understand why a system is built a certain way, the rationale behind the design must be preserved for future designers and developers. However, it is sometimes difficult to record all the rationale into a document. Often, this rationale is not discussed anywhere outside of the design meetings. The Software Architecture Analysis Method (SAAM) was developed at the Software Engineering Institute in the mid to late 1990's to support the organized discussion of architectural rationale [16]. SAAM is a people-oriented process and centers around group meetings where people produce scenarios to help extract how changing requirements impact an already existing system. A typical SAAM session is a live event involving discussions by 3-10 of the stakeholders involved in the system including users, designers, managers and facilitators that is centered around drawings of the architecture on a public display.

2.3.2 How It Works

We prototyped a capture and access system to support the capture of SAAM sessions (see Figure 2). Both the architectural diagrams and the discussions around these diagrams provide the rationale behind the architecture of a system. The diagrams and discussions are, therefore, important aspects of the meetings that we want to capture and relate later on. By converting the public display to an electronic whiteboard surface and recording the discussion with digital streaming media technology, we were able to capture the SAAM sessions and then provide the ability to salvage summary information afterwards.

The primary capture application was designed to capture the architectural drawings generated during these meetings. As the participants discussed the specifics details about a component, the application stored that part of the discussion under the appropriate architectural block. When each scenario is presented, both the architectural diagrams and discussions that come about are captured. After all the scenarios are captured, they can be reviewed to assess the implications and the rationale of changes to system to determine how to evolve the system properly.

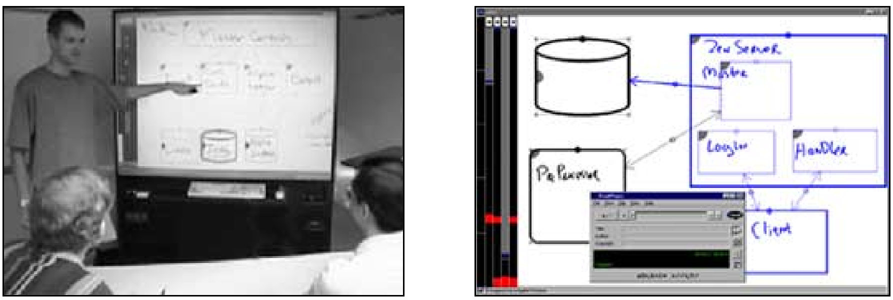

Fig. 2 Software Architecture Analysis Method Pad. The left shows stakeholders participating in the capture of a software design meeting. The right shows the access of a SAAM session

2.3.3 Related Work

Some capture and access systems used in meetings, such as the Marquee [33] and the Filochat [35] are private note-taking systems and do not capture a shared group perspective of the discussion – rather, they capture the individual note-taker's view. However, in some meetings, particularly in design meetings, it is more important to have the shared understanding of how a system is designed (or how it is agreed to be designed) be preserved for posterity. SAAMPad captures the evolution scenarios presented by a stakeholder.

Tivoli [24] draws the most similarity to the SAAMPad system. It introduces the notion of "domain objects" to represent very specific kinds of artifacts that are captured. In order to capture very specific kinds of artifacts (such as software architecture diagrams in SAAMPad, or Intellectual Property (IP) documents in Tivoli), these applications rely on knowledge of the well-defined formal aspects of the live experience.

2.4 DUMMBO (Dynamic, Ubiquitous, Mobile Meeting Board)

2.4.1 Motivation

Not all meetings can be scheduled ahead of time, resulting only from serendipitous encounters. The subject of what people talk about during these opportunistic meetings is not known ahead time, nor is it known in advance who is involved, or even how long the meeting will last. Public whiteboards are the site of where a lot of these types of informal meetings take place. These boards are often placed in locations where there is a reasonable flow of traffic to encourage anyone who passes to discuss ideas and to brainstorm with one another. However, it is difficult to know when someone is involved in an informal discussion at the whiteboard versus just doodling on it. This presents a challenge for how to support the unintrusive capture of informal meetings in a way that facilitates the retrieval of the conversation and writings on the whiteboard.

2.4.2 How It Works

We created a system called DUMMBO [6] (see Figure 3), using a non-projecting SmartBoard with an attached sound system, to capture informal and opportunistic meetings. When anyone approaches the board and picks up a pen to write, the board automatically begins to capture the writing and discussion. After a certain period of inactivity, recording will stop. Sensing technology is instrumented near the whiteboard to detect the people present during each meeting. If two or more people are known to be near the board, then recording of the conversation will occur even if no writing appears on the electronic whiteboard. A Web interface is provided to support the access of this collection of unstructured meetings. The context of an informal meeting (who was there, when and where it occurred) is used to help an individual find a meeting of interest. Users may browse through a timeline displaying periods of activity at the board and may apply filters (who, where, when) to pinpoint a meeting of interest. Once an appropriate time period has been selected, and the correct meeting has been retrieved, the access interface allows the user to replay the whiteboard activity, synchronized with the audio.

2.4.3 Related Work

Xerox PARC's Flatland project [22] has also looked at the capture and access of informal activities, but it uses time as the only mechanism used to retrieve historical information. Flatland was designed to support informal activities within a private office. DUMMBO is intended to support the capture of informal activities on a public whiteboard where many people can interact with it at any given time. For the situation that DUMMBO is intended to support, time alone is not good enough to help users forage through a potentially large set of information to find just their discussion.

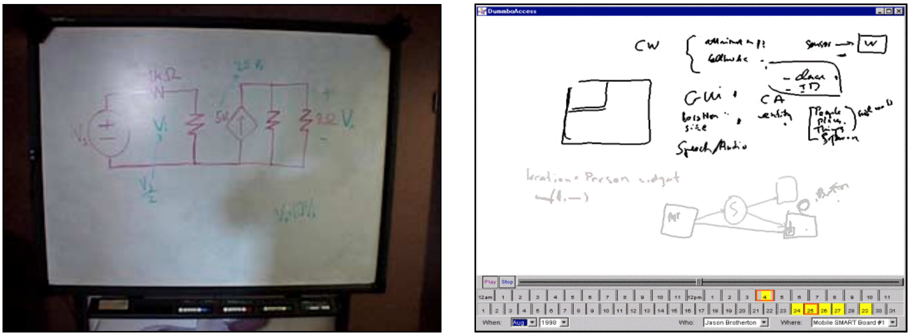

Fig 3 DUMMBO. The left shows the instrumented electronic whiteboard that sits in a public space. The right shows the context-based access interface

2.5 The Conference Assistant

2.5.1 Motivation

Large academic conferences often have multiple tracks of concurrent activities. Paper presentations, demonstrations, special interest group meetings, etc., can all occur simultaneously. During any given time period, conference attendees can move about and listen to any of the presentations of relevant interest. To help remember what was seen, attendees can take notes to document the sessions they attended. The abundance of potentially novel and interesting information presents the attendees with a conflict between taking notes and synthesizing the sessions. Furthermore, the large amount of information, ranging over many different topics makes it difficult to organize the notes.

2.5.2 How It Works

The Conference Assistant [11] is a mobile capture and access application that allows users to take notes at distributed presentations. As attendees arrive at a conference, they are a given a handheld PDA for use during the conference. Rather than needing to take detailed notes, presentations are captured by the rooms. As a result, conference attendees can take summary notes on the personal and mobile devices. After the conference concludes, each attendee’s notes can be integrated with the presentations.

When attendees move about between presentations, their mobility presents a challenge for integrating each user’s private notes with the appropriate presentation they had seen. The Conference Assistant is not only a capture and access application but is also a context-aware application, logging the physical location of the user at all time (through use of RF ID positioning technology). As presentations are given, they are captured and tagged as given in certain locations. After the conference ends, when an attendee reviews a talk she attended, the location information is used to integrate the personal notes with the actual presentation information.

2.5.3 Related Work

The notion of context-based retrieval is not novel, but rather is seldom used in the domain of capture and access applications. Most applications use time as the integrating variable between various streams of information. However, when multiple capture-enabled environments are available and users can move between any of these physical spaces, more information beyond time is needed to integrate the appropriate information. DUMMBO, described earlier, uses time, location, as well as the identities of the people present during the live experience to retrieve the appropriate set of information. Systems such as StuPad, described earlier, and NoteLook [9] allow the personalization of captured information in settings such as classrooms or seminars but assume fixed location. NotePals (which also has been used at conferences as well as classrooms), like the Conference Assistance, integrates the notes based on matches in context, such as location, etc.

2.6 TeamSpace

2.6.1 Motivation

Project group meetings are used to discuss various aspects of the group project. Meetings can be devoted to understanding the team's progress; specifics on how important parts of the projects are implemented (or will be implemented) are sometimes presented, agendas are drawn out, and schedules and responsibilities are defined. Traditionally, these meetings involve multiple people who come together at a mutual location. As companies look to grow world wide, the nature of the work place is now a distributed environment with multiple people at different geographical locations collaborating in a large project. The distributed work environment means that these meetings now occur through a virtual connection while people remain scattered in multiple (and often remote) locations.

2.6.2 How It Works

The TeamSpace project supports the capture of these meetings as multimedia meeting notes as part of a larger set of shared artifacts created and maintained for each project [25]. Meetings are typically held in a number of meeting rooms and/or offices. A capture application was built to capture and share streams of information between the different instances of it (running at the different sites involved in the meeting). The different streams of information the application supports include presentation slides, annotations, agenda items, action items, and video frames. Telephone connections are used to provide an audio connection between these physical spaces. Thus audio is captured through the phone line, although potentially a voice over IP solution could be instrumented as well.

2.6.3 Related Work

In comparison to other meeting capture and access applications, TeamSpace can be used to capture a single collocated collaborative meeting (such as Tivoli, etc.). However, TeamSpace allows for multiple people to collaborate in the capture of the streams of information (similar to DOLPHIN [30]). Furthermore, it goes beyond just allowing multiple devices to control a single meeting surface (such as Pebbles [21]). More compellingly, TeamSpace provides multiple people and multiple locations with the chance to participate in the capture of information. The key difference is everyone can capture information and it must be shared across all location.

3 Design Space of Capture & Access Applications

The designs of the six applications presented above, as well as those applications found in the literature, face a variety of issues. The minimal set of issues that need to be addressed can be extracted when five design dimensions across two phases relevant to all capture and access applications are taken into consideration in the design. For each of these applications, the *capture* phase is typically the live experience duration to be preserved, while the *access* phase happens when these records are reviewed. For each of these phases, designers must determine:

- *Who* are the users during capture and access?
- *What* is captured and accessed?
- *When* does capture and access occur?
- *Where* does capture and access occur?
- *How* is capture and access performed?

3.1 The Who Dimension

Building a capture and access application is like building any kind of application: it is extremely important to identify *who* the users are. The *who* component of a capture and access application deals with the scale of users and the users' roles. For different kinds of situations, there may be a different number of users participating in that experience. Furthermore, the people involved during the capture of an experience do not necessarily have to be those who will be accessing the information. In some situations, even though the people present during the live experience are the same set of people who will access the captured information, they may not all directly participate in the capture of the experience. Each person present during the capture or the access of the experience may have a part in the experience. In understanding each person's part, designers can design systems to support specific roles in the capture and access of the experience. As Grudin points out, much of the success of groupware systems depend on who amongst the participants are actually doing the work and who can directly benefit from the work [14]. There is much similarity between groupware and capture and access applications. Beyond recognizing who are the users, designers should also design applications to take advantage of people's roles in the experience. When a single user who plays a major role in an experience (such as an instructor during class) can capture information on behalf of all those present (the students) to review in the future, there is clear benefit for the students to have an automated capture system developed around the activities of the instructor.

3.2 The What Dimension

There are many everyday activities that can be aided by the services provided by capture and access applications. However, each live experience may vary in the amount of pre-defined structure. Identifying the formal characteristics of the experience helps identify the parts of the experience that can be captured and how to design an application to record this set of information. Once data has been captured, users can review the live experience. However, the live experience can never be fully recorded. While the actual experience sets the ceiling for what is captured, the amount of information actually captured sets the ceiling for the access of the experience. To increase the fidelity of the access experience, more streams can be captured and integrated; collectively, they can give a more accurate the account of the experience. The *what* dimension is defined by the number of streams of information captured for later access. One approach is to set the scale as high as possible by capturing as much information as possible. Information can always be processed and filtered; however, when uncaptured information is lost to the past, there is no way to recreate it. On the other hand, information is captured so that it can be accessed in the future – capture is meaningless without access. Whereas it is important to make the live capture as complete as possible, there is also no value in doing the capture if there is no reasonable and useful access to the captured record. Hence, the needs of access should not be ignored. If a low fidelity of access is all that is ever going to be required, then there is no need to capture more information than what is needed. The tradeoff between the two approaches is the effort used to design a system to capture as much as possible versus the effort used to understand the access needs for that system.

3.3 The When Dimension

The *when* dimension deals with issues related to when capture occurs, when access occurs, and the time difference between the capture and access activity. Knowing when capture occurs means devices and applications can be designed to take advantage of this known piece of context to infer others – predicting and customizing the right kinds of capture services to initiate automatically for unique situations. Some experiences occur periodically or with some frequency. However, when experiences occur unplanned or on irregular intervals, applications must be ready for capture at any point and flexible enough to adapt to the changing requirements of the situational context. Using knowledge of when access occurs can also inform the design of the access devices and applications. Knowledge of when access occurs can be used to provide the users with the right applications for the right available resources. Not knowing this information means "anytime" access support must be provided, where the context of the access experience is unknown and applications must be built to support impoverish resources.

The time difference between the capture and access activity can also be used to inform the design of the applications. Short-term capture and access applications are typically used as reminder systems. Xcapture [15] and Where-Were-We [17] are examples of near-term capture and access applications, capturing either audio or video of an activity and allowing the users to index into the captured streams when they need to be reminded of certain pieces of information during an activity. When the access of information occurs only immediately after it has been captured (or within a day or two), persistent storage of the information may not be necessary. Medium-term applications typically have data accessed in the weeks to months range of time after it has been captured. These applications will need to physically store the information and furthermore can process and transduce the information into friendlier formats than the raw data that is captured. Finally, long-term applications store information as records for posterity. Information needs to persist for much longer periods of time than other types of applications and it may make sense to provide users with a synthesized summary of the experience with an interface that supports being able to drill down to the exact point that the user(s) want to review.

3.4 The Where Dimension

The *where* dimension addresses the physical locations involved in a capture and access phases. Most capture and access applications handle experiences that occur in a single location. However, it is becoming more commonplace for people in many different places to collaborate and essentially share an experience remotely. Furthermore, capture and access applications must also take user mobility into consideration.

Knowing where capture occurs means being able to instrument the physical space involved with capture devices and applications ahead of time. Identifying where access occurs can help inform the design of the access applications. Knowledge of where people will want to access information will provide understanding to the resources they have available – what kind of machines the users have available to them, the kind of input and output capabilities they have, etc. If access occurs in the same environment as the live experience (i.e. there is an overlap between where capture and access occurs) specialized devices can be instrumented to be both capture as well as access tools.

3.5 The How Dimension

The tools and methods for capturing and accessing information as well as the scale of devices form the last dimension: *how*. Capture and access applications are typically built as a confederation of tools. The number of devices that are used in a system defines the scale of devices for capture and access applications. At one end of the scale, only a single device is used in the application. A key question in the building of capture and access devices is whether the device that is doing the capture can also be used to provide the access. In some instances, personal and portable devices play dual roles as both the capture and the access devices. When a device does not support the access of its captured content, or is not the best tool to use when accessing the captured experience, there is a need to identify what are the other devices that support access to the information. In most cases, capture is often done using a number of devices and so a certain amount of effort must be devoted to coordinating these devices to work together. If the tools users work with can be identified ahead of time, designers can augment these devices with capture or access capabilities. Devices and applications can be instrumented to support the explicit interaction of the users to capture information or they can be fully automated to support capture. Explicit capture of experiences is more common, but requires effort on the part of the users. We can ease the explicit production method by automating the capture of materials – making the capture implicit. Fully implicit capture applications can make capture completely transparent to the user and can be done in the background. However, from the designer's point of view, these systems will require some level of intelligence. For example, systems that rely solely on cameras to capture will also need vision techniques to detect changes in the material that is being captured. Systems that rely on explicit interactions can be instrumented to capture at a higher level of granularity. The tradeoff between these two approaches lies in the granularity of capture as well as the level of intelligence built in to the capture systems. By instrumenting the physical objects the user interacts with, we are able to obtain a finer level of granularity in the interaction history without needing to apply much intelligence into the system.

4 Conclusion

The five dimensions we have outlined, as well as the list of key attributes for each dimensions, are not exhaustive. Clearly, privacy and security are among the list of those issues not mentioned here. These issues are certainly not lesser in importance; in fact, these issues may be the more challenging ones to explore. Instead the issues we have outlined here are what we believe to be common ones that need to be considered when building *any* capture and access application. As a result, these five dimensions should always be addressed at the start of the design of a capture and access application. By identifying the *who*, *what*, *when*, *where*, and *how* components of a capture and access application, it is possible to extract the functional requirements in the development of a system.

In generalizing a design space for capture and access applications, we examined the issues involved in the design of not only the capture and access applications we have created but those of related work found in the literature. When we assess both our own work and other's with respect to the design space, we can see the parts of this space that existing applications populate, as well as the holes in this space that remain to be explored. Few projects have explored the following issues:

- Long-term access of captured data
- Capture of informal experiences
- Capture of distributed and remote experiences
- Capture mobility
- The personalization of capture
- Context base (context rich) capture and access
- Instrumentation of dual role devices
- More variety in the types of devices that are augmented with capture and access capabilities

In future work, we will present a prototype for an infrastructure to capture and access (known as InCA – Infrastructure for Capture and Access) which will look at how the task of building a capture and access system can be facilitated through lower level support that includes reusable components that form building blocks for the capture, storage, and subsequent access of the experiences. We identify five dimensions for capture and access in this paper. InCA aims to support developers in the development of any application ranging anywhere on these dimensions. By making it easier to build systems that exists in this design space, application developers will be able to build more compelling capture and access applications in the future. Furthermore, they will be able to explore issues that have been previously unvisited, such as those listed above.

Acknowledgements

The authors are members of the Future Computing Environments (FCE) group at Georgia Tech. We would like to thank all the visiting researchers and other members of the FCE group who have helped to shape this paper. We would like to acknowledge Heather Richter and Anind Dey for their contributions to some of the projects discussed in this paper. The TeamSpace project discussed in this paper is joint research performed between Georgia Tech, IBM Research and Boeing. Other research projects discussed are funded through various organizations (including the National Science Foundation, DARPA, and the Army Research Lab) and a number of industrial sponsors, including the members of the Aware Home Research Initiative, Sun Microsystems and Hewlett-Packard.

References

1. Abowd, G. D., Classroom 2000: An experiment with the instrumentation of a living educational environment. IBM Systems Journal. **38**(4) (1999) 508-530
2. Bacher, C., et al. Authoring on the Fly. A New Way of Integrating Telepresentation and Courseware Production. In *Proceedings of ICCE 1997.* Kuching, Sarawak, Malaysia (1997)
3. Berque, D. Using a Variation of the WYSIWIS Shared Drawing Surface Paradigm to Support Electronic Classrooms. In *Proceedings of HCI International 1999.* Munich, Germany (1999)
4. Bianchi, M. AutoAuditorium: A Fully Automatic, Multi-Camera System to Televise Auditorium Presentations. In *Proceedings of DARPA/NIST Smart Spaces Technology Workshop.* (1998)
5. Black, M. J., et al. The Digital Office: Overview. In Proceedings of 1998 AAAI Spring Symposium on Intelligent Environments. (1998)
6. Brotherton, J. A., Abowd, G. D., and Truong, K. N. Supporting Capture and Access Interfaces for Informal and Opportunistic Meetings. Georgia Institute of Technology Technical Report GIT-GVU-99-06 (1998)
7. Brotherton, J. A., Bhalodia, J. R., and Abowd, G. D. Automated Capture, Integration, and Visualization of Multiple Media Streams. In *Proceedings of IEEE Multimedia and Computing Systems.* Austin, TX (1998)
8. Bush, V.: As We May Think, in *Atlantic Monthly*. (1945)

9. Chiu, P., et al. NoteLook: Taking Notes in Meetings with Digital Video and Ink. In *Proceedings of ACM Multimedia 1999.* Orlando, FL (1999) 149-158
10. Davis, R. C., et al. NotePals: Lightweight Note Sharing by the Group, for the Group. In *Proceedings of CHI 1999.* Pittsburgh, PA (1999) 338-345
11. Dey, A. K., et al. The Conference Assistant: Combining Context-Awareness with Wearable Computing. In *Proceedings of ISWC 1999.* San Francisco, CA (1999) 21-28
12. Elrod, S., et al. Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of CHI 1992.* Monterey, CA (1992) 599-607
13. Ginsberg, A. and Ahuja, S. Automating envisionment of virtual meeting room histories. In *Proceedings of ACM Multimedia 1995.* San Francisco, CA (1995) 65-76
14. Grudin, J.: Groupware and Social Dynamics: Eight Challenges for Developers, in *Communications of the ACM*. (1994). p. 92-105
15. Hindus, D. and Schmandt, C. Ubiquitous Audio: Capturing Spontaneous Collaboration. In *Proceedings of Computer Supported Collaborative Work 1992.* Toronto, Canada (1992) 210-217
16. Kazman, R., et al. SAAM: A Method for Analyzing the Properties of Software Architectures. In *Proceedings of International Conference on Software Engineering (ICSE 16).* (1994) 81-90
17. Minneman, S. and Harrison, S. Where were we: Making and using near-synchronous, pre-narrative video. In *Proceedings of ACM Multimedia 1993.* (1993) 1-6
18. Minneman, S., et al. A confederation of tools for capturing and acessing collaborative activity. In *Proceedings of ACM Multimedia 1995.* San Francisco, CA (1995) 523-534
19. Moran, T. P., et al. "I'll Get That off the Audio": A Case Study of Salvaging Multimedia Meeting Records. In *Proceedings of CHI 1997.* Atlanta, GA (1997) 202-209
20. Mukhopadhyay, S. and Smith, B. Passive Capture and Structuring of Lectures. In *Proceedings of ACM Multimedia 1999.* Orlando, FL (1999) 477-487
21. Myers, B. A., Stiel, H., and Gargiulo, R. Collaboration Using Multiple PDAs Connected to a PC. In *Proceedings of Computer Supported Cooperative Work 1998.* Seattle, WA (1998) 285-294
22. Mynatt, E. D., et al. Flatland: New Dimensions in Office Whiteboards. In *Proceedings of CHI 1999.* Pittsburgh, PA (1999) 346-353
23. Newman, W. and Wellner, P. A Desk Supporting Computer-based Interaction with Paper Documents. In *Proceedings of CHI 1992.* Monterey, CA (1992) 587-592
24. Pedersen, E. R., et al. Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. In *Proceedings of ACM INTERCHI 1993.* Amsterdam, The Netherlands (1993) 391-398
25. Richter, H., et al. Integrating Meeting Capture within a Collaborative Team Environment. In *Proceedings of UbiComp 2001.* Atlanta, GA (2001)

26. Richter, H., Schuchhard, P., and Abowd, G. D. Automated capture and retrieval of architectural rationale. Georgia Institute of Technology Technical Report GIT-GVU-98-37 (1999)
27. Smart Technologies, Inc. SmartBoard. http://www.smarttech.com.
28. Stafford-Fraser, Q., et al. BrightBoard: A Video-Augmented Environment. In *Proceedings of CHI 1996.* Vancouver, Canada (1996) 134-141
29. Stifelman, L. J. The Audio Notebook.Ph.D. Thesis, Media Laboratory, MIT (1997)
30. Streitz, N. A., et al. DOLPHIN: Integrated Meeting Support across Liveboards, Local and Remote Desktop Environments. In *Proceedings of Computer Supported Collaborative Work 1994.* Chapel Hill, NC (1994) 345-357
31. Truong, K. N., Abowd, G. D., and Brotherton, J. A. Personalizing the Capture of Public Experiences. In *Proceedings of UIST 1999.* Asheville, NC (1999) 121-130
32. Wactlar, H., et al. Informedia Experience-on-Demand, Capturing, Integrating and Communication Experience across People, Time and Space. http://www.informedia.cs.cmu.edu/eod.
33. Weber, K. and Poon, A. Marquee: A tool for real-time video logging. In *Proceedings of CHI 1994.* Boston, MA (1994) 58-64
34. Weiser, M., The computing for the 21st century. Scientific American. **265**(3) (1991) 94-104
35. Whittaker, S., Hyland, P., and Wiley, M. Filochat: Handwritten notes provide access to recorded conversations. In *Proceedings of CHI 1994.* Boston, MA (1994) 271-277
36. Wilcox, L., Schilit, B. N., and Sawhney, N. Dynomite: A Dynamically Organized Ink and Audio Notebook. In *Proceedings of CHI 1997.* Atlanta, GA (1997) 186-193