UNIVERSITY OF CALGARY

Artifact Awareness for Distributed Groups through Screen Sharing

by

Kimberly Eleanor Tee

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

DECEMBER, 2007

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled “Artifact Awareness for Distributed Groups through Screen Sharing” submitted by Kimberly Eleanor Tee in partial fulfillment of the requirements for the degree of Master of Science.

Supervisor, Dr. Saul Greenberg
Department of Computer Science

Dr. Jonathan Sillito
Department of Computer Science

Dr. Ehud Sharlin
Department of Computer Science

Dr. Edna Einsiedel
Faculty of Communication and Culture

Date

Abstract

When co-located, people can see the artifacts that others are working on, which in turn enables casual interactions. To investigate whether we can help distributed groups maintain mutual awareness of people’s electronic work artifacts, I designed and implemented an awareness tool that uses screen sharing. People see portions of others' screens in miniature, can selectively raise larger views of a screen to get more detail, and can engage in remote pointing. People balance awareness with privacy by using several privacy-protection strategies built into the system. An evaluation with two groups using this system shows that people use it to: maintain awareness of what others are doing, project a certain image of themselves, monitor progress, coordinate joint tasks, determine others’ availability, and engage in serendipitous conversation and collaboration. While privacy was not a large concern for these groups, a theoretical analysis suggests that privacy risks may differ for other user communities.

Publications

Some of the materials, ideas, and figures in this thesis have previously appeared in the following publications:

Tee, K., Greenberg, S., and Gutwin, C. (2006). **Providing artifact awareness to a distributed group through screen sharing**. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2006). ACM Press, 99-108.

Tee, K., Greenberg, S., Gutwin, C., and McEwan, G. (2006). **Shared desktop media item: The video**. In Video Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2006). ACM Press, Video and two-page summary.

Acknowledgements

There are many people who have supported and encouraged me throughout this research that I would like to acknowledge.

First and foremost, I would like to thank my supervisor, Dr. Saul Greenberg: Thank you so much for your guidance, encouragement, support, and patience over the past few years. You have provided me with so many opportunities I otherwise wouldn't have had, and I feel incredibly lucky to have been one of your students.

Dr. Jonathan Sillito, Dr. Ehud Sharlin, and Dr. Edna Einsiedel: Thank you for your insightful comments and questions about my work, and for the time and effort you spent on examining it.

Gregor McEwan: Thank you for helping me out so much with Community Bar – answering all my questions, changing the implementation to support some of the things I wanted to try, and basically being willing to give help anytime it was needed.

Mark Hancock: Thank you so much for all your advice and encouragement, especially during this past year. Without your help and support, things would not have gone as well as they did, or been as much fun as they were. Thank you for always being there for me, both in the lab and out of it (i.e., both on the ultimate field and off of it!).

To my friends and colleagues from the Interactions Lab: I am very grateful for all your support. I have learned so much from you all, and you have helped me to become a better researcher and presenter. In particular, I'd like to acknowledge the Maple Story players and the squash group for making things especially fun during this time!

To my family, especially my parents, Bill and Marife Tee: Thank you for encouraging me and believing in me, even when it seemed I would be in school forever.

Finally, I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC), Alberta's Informatics Circle of Research Excellence (iCORE), and the Alberta Ingenuity Fund (AIF) for their financial support.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

In this thesis, I focus on how to help co-located and distributed members of a small group maintain awareness of each others' work artifacts (as visible on their computer screens), where they use this awareness to find opportunities for conversation and collaboration with each other. In this first chapter, I introduce the context and motivation behind my research. I describe some of the challenges that co-located and distributed group members face in maintaining awareness of each others' work, and I briefly overview prior research done on groupware systems for supporting awareness and interaction between members of a small group. Next, I state the research questions this thesis investigates and describe my approach to addressing them. Finally, I conclude this chapter with an organizational outline of the remainder of this thesis.

1.1 Context and Motivation

Previous studies have shown that *casual interaction* – the brief, unplanned meetings that commonly occur during the day between co-located people – is important for coordinating joint work, tracking progress of joint work, exchanging knowledge and information, and building relationships (Kraut et al., 1988; Whittaker et al., 1994). Casual interaction is made possible by *informal awareness*, the naturally gained understanding of who is around, what tasks they are performing, and whether or not they are available for conversation or collaboration (Kraut et al., 1988).

Informal awareness is easy to maintain in a co-located setting, such as when people inhabit a shared office space. Just by being in the same environment, people naturally accumulate background information about what is going on around them (Bly et al., 1993). People also do *walkbaouts*, where they wander around the shared space just to see what others are up to (Bellotti and Bly, 1996). Because many awareness cues are available in a co-located environment, transitioning to casual interaction is typically effortless.

For distributed groups, however, initiating casual interaction is problematic: people do not see who is around, do not know if others are available for conversation, and lack the awareness cues that naturally lead to serendipitous interaction. Consequently, distributed collaborators must expend a relatively large amount of effort to explicitly coordinate interaction (e.g., by scheduling meetings), or do without this interaction altogether. Kraut et al. (1988) argue that much useful communication between workers in a knowledge-based environment is unplanned and would not occur if it had to be planned, suggesting that distributed groups are missing out on valuable interaction opportunities that naturally occur in co-located groups. This partially explains the explosion and success of low-effort awareness servers and casual interaction systems, such as the widespread adoption of instant messengers by diverse user groups (Nardi et al., 2000), or how email is often used for casual on-going conversations rather than as a formal messaging system. Many existing tools succeed because: (a) they let people know about the activities and thus the approximate availability of their colleagues, and (b) they make initiating conversation extremely easy.

Yet, an important component of informal awareness that is not handled by these awareness servers and instant messengers – and the focus of this thesis – is *artifact awareness.* I define artifact awareness as:

> *one person's up-to-the-moment knowledge of the artifacts and tools that other people are using as they do their work.*

For office workers, artifacts include the documents and drawings (both physical and digital) that people work on over the course of a day, the secondary materials that support their tasks, and the tools they use to carry out their work.

Artifact awareness is typically easy when collaborators are in a co-located environment because people naturally gather visual and auditory cues about other people's presence and activities. They notice what artifacts others are working on as they glance into people’s offices, look at their desks, and see their computer displays. In the real world, for example, if a document on a person’s desk catches the eye of someone walking by, that person can stop and discuss that artifact with its owner. This awareness can be very important for exchanging knowledge and building social relationships.

For example, consider designers who rely on easy visibility of other people’s work. Design groups intentionally work in studio spaces to promote learning, reflection, and discussion about current projects – designs are placed on easels or other semi-public surfaces, and others moving through the studio can monitor and comment on the work as it unfolds over time (Buxton et al., 2000). Similarly, many command and control situations – such as air traffic control, subway routing, or shipboard navigation – rely on people being able to see each other’s artifacts (Heath and Luff, 1992). Artifact visibility also plays a role in how people create common ground in conversations (Clark, 1996).

However, even in a co-located environment, artifact awareness can be difficult to maintain, particularly if the artifact is digital. Most digital artifacts only appear on a screen, which hinders awareness when people’s bodies shield their on-screen work from others or when social etiquette prevents others from taking a closer look at someone’s display. Part of the etiquette problem is that the screen contains both semi-public information and highly personal information. A viewer cannot distinguish between these unless he or she is looking closely at the information, and by that time, it is too late to avoid looking at any private information. Another problem with maintaining awareness of digital artifacts is that with current computer windowing systems, artifacts are often overlapped or are not even shown on screen when another artifact is being used. Switching from using one digital artifact to another digital artifact takes little time and can be easily missed by others.

Artifact awareness is even more difficult to maintain in a distributed setting; despite the availability of awareness servers and casual interaction systems, distributed groups still lack the easy awareness of others’ artifacts that is normally found in a co-located shared environment. Whittaker et al. (1994) found in their study that a little over half of all casual interactions in an office involved document sharing, strongly suggesting that distributed groups would benefit from being able to more easily share artifacts. Yet, for distributed groups, there is still no real equivalent to the way co-located people can visually share their individual work and maintain artifact awareness. While there are many groupware systems (e.g., shared editors and webcast meeting tools) that do let a distributed group share artifacts, they work only *after* interaction is initiated. That is, they are intended for focused collaborative work rather than for artifact awareness. They also tend to be heavyweight to set up, e.g., they may involve many interaction steps to get going, which inhibits their use for casual interaction.

Perhaps the closest equivalent for distributed groups are *screen sharing* applications that let people explicitly share their computer screens, windows, or window fragments with each other. Screen sharing, first demonstrated in 1968 (Engelbart and English, 1968), has been used in practice for over twenty years in real-time distributed collaboration systems, e.g., Farallon Timbuktu (WOS Data Systems, 1987) and VNC (Richardson et al., 1998). It is often a key component of desktop conferencing systems, in which audio/video teleconferencing technologies are integrated with desktop computer application sharing in order to allow individuals to meet, collaborate and work together from their offices. Other common uses of screen sharing include application sharing (desktop conferencing without requiring the use of audio/video) and remote assistance (a system administrator or an expert user can remotely control another's computer to assist them in performing certain tasks). While useful, all these systems are designed for focused interaction rather than artifact awareness leading to casual interaction.

However, I hypothesize that the concept of screen sharing can be used to support artifact awareness and opportunistic interaction in a small work group where members have a strong desire to stay in up-to-the-moment contact. While the focus in this thesis is on distributed groups, I hypothesize that screen sharing can also be used to support artifact awareness and opportunistic interaction between people in a co-located group. The assumption is that much of each group member's work is done on their computer, so that by using screen sharing, almost all of their activities can be captured and shared to provide the group with mutual awareness of people's artifacts and individual work.

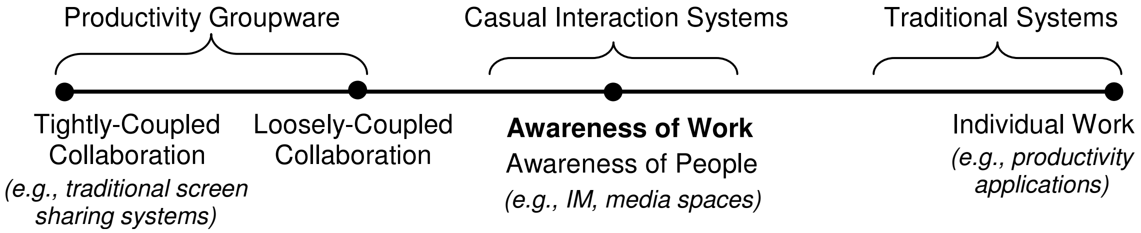

Productivity Groupware
Casual Interaction Systems
Traditional Systems
Tightly-Coupled Collaboration
(e.g., traditional screen sharing systems)
Loosely-Coupled Collaboration
Awareness of Work
Awareness of People
(e.g., IM, media spaces)
Individual Work
(e.g., productivity applications)

Figure 1.1 – The focus of my thesis is to provide awareness of work between members of a distributed group.

To put this hypothesis into context, consider Figure 1.1. The left side covers traditional screen sharing, desktop conferencing, and teleconferencing systems, where people engage in a meeting session to do tightly-coupled work. The right side emphasizes traditional single user systems as embodied in most computer applications. The middle region is this thesis' focus – systems that support causal interaction. Prior research that fits into this middle region includes research on instant messaging (IM) and media space systems, as both provide informal awareness and casual interaction to distributed groups. However, awareness in these systems is usually centered around the state of people – awareness of work leading to casual interaction is largely unsupported.

IM systems, which have recently become extremely popular both at home and in the workplace, generally provide *status indicators* when people are logged on showing whether they are busy or available for conversation. Some systems also show whether a user is engaged in a particular activity, such as a chat with another user. Initiating conversation with another person typically involves clicking on their name. While IM systems enable people to quickly communicate with each other using little effort, the awareness information they provide is very basic, making it difficult to know what others are doing or what they are working on.

Media spaces provide rich awareness by linking offices and public spaces through networks of audio and video. Through video, viewers can easily see who is around and what they are doing. However, their actual work artifacts are rarely visible due to camera angle and resolution limitations (although some early systems – notably Cruiser (Root, 1988) – did have a second camera that could be pointed at the desk). Some recent media spaces also allow group members to augment videos by posting media elements, e.g., editable sticky notes or web page thumbnails. These contribute to public conversations and shared information in a real-time area that all members can see. However, they do not provide awareness of others' actual work, unless this information is explicitly offered during conversation or in a broadcast message.

To date, little work has been done on using screen sharing for awareness (Figure 1.1, middle). One exception is SynchronEyes [www.smarttech.com], a commercial system that lets one person view others' desktops as thumbnails. However, it is designed for monitoring (vs. a peripheral awareness tool for casual interaction), in a quite different educational setting where a single teacher monitors / controls a class of students (i.e., many-to-one). Another exception is the Multi-VNC system (Gutwin et al., 2005), a proof-of-concept prototype in which multiple instances of an open-source screen-sharing application were run so that each member of a group could see the other members' computer desktops. As well, the Notification Collage Desktop media item (Rounding, 2004) showed auto-updating desktop thumbnails for the people who posted them and could be used to activate a full screen sharing session, but was built only to show that third-party software could be easily linked into the system. These latter two research projects have a direct lineage with my own work as they involve collaborations with my thesis supervisor. These projects are precursors that motivate my research, as neither examined in detail how to provide awareness using screen sharing.

1.2 Research Questions

In this thesis, I investigate the use of screen sharing for supporting artifact awareness and opportunistic interaction between *intimate collaborators* – people with a strong need or desire for close coordination, collaboration, or social interaction. While the focus in this thesis is on distributed groups, I also consider co-located groups. In particular, I address the following research questions.

1. *How can awareness of ongoing individual work be increased between intimate collaborators, particularly members of a distributed group?* Existing informal awareness and casual interaction tools generally only provide information about

who is around and what they are talking about. A few also offer the ability to post (usually static) artifacts for others to view, such as photos and web links of public interest. Few systems share what people are working on (i.e., their individual work) or what they are interested in, without them having to explicitly tell others or move to a "meeting" mode.

2. *What mechanisms can help people control how much information they reveal to others? What mechanisms can let others control how much information they receive?* For someone sharing their individual work with others, the challenge for an awareness tool is to provide information about that person's work / activities while still meeting that person's privacy needs. Clearly, someone opting to share his or her artifacts and work with others will want to somehow monitor and control what is being shown. Yet, it is uncertain how this should be done. Similarly, an awareness system must also consider that people receiving this information may want access to varying amounts of awareness information in different situations. That is, someone receiving shared information should be able to control how much they can see, particularly if there is information they are not interested in or that they find distracting. Again, it is uncertain how this should be done.

3. *How do such systems work in practice?* While there have been prior studies of how existing informal awareness and casual interaction tools are used by various groups, existing tools generally do not provide awareness of ongoing individual work. Consequently, we do not know how such an awareness system would be adopted and used in practice.

1.3 Approach

In this thesis, I document the design, implementation, and evaluation of a screen-sharing awareness tool used to address each of the stated research questions.

1. *Screen-sharing awareness tool.* I researched existing informal awareness and casual interaction tools to inform the design and implementation of an awareness tool that uses screen sharing to support artifact awareness between intimate collaborators.

This awareness tool was built within the context of the Community Bar (McEwan and Greenberg, 2005), a media space that can be augmented with extra information.

2. *Privacy controls and feedback; different levels of information and interaction*. I incorporated several privacy controls and feedback mechanisms into the screen-sharing awareness tool for people sharing their screens, then evaluated these controls and feedback mechanisms by applying Boyle's privacy framework (Boyle, 2005) to the awareness tool in a theoretical analysis of privacy concerns. Additionally, I included various levels of information and interaction possibilities for viewers to see shared artifacts and screens at different levels of detail.

3. *Evaluation with end-users*. I collected qualitative data from two different groups on how they each used the screen-sharing awareness tool over a two-week period. One group was an internal research lab that had already been using Community Bar on a daily basis for over a year. The other group was an external commercial development team, which had been introduced to Community Bar and the screen-sharing awareness tool at the same time.

1.4 Thesis Overview

The remainder of this thesis describes in detail the research outlined above. In Chapter 2, I discuss the background and motivation for this work. I describe informal awareness and casual interaction, and overview some of the current research in this area. I also overview related work done on screen sharing. In Chapter 3, I present several scenarios showing how screen sharing for awareness can be beneficial or harmful, depending on how it is used. Then, I describe the screen-sharing awareness tool and the design rationale behind it. In Chapter 4, I describe the architecture and implementation details, as well as report on the results of performance tests done on the screen-sharing awareness tool. In Chapter 5, I report on an evaluation involving two different groups of intimate collaborators. I discuss their initial experiences and reactions, and highlight common findings between the two groups. In Chapter 6, I present a theoretical analysis of how the tool can affect its users' privacy, including an articulation of possible privacy concerns. I also discuss two hypothetical scenarios of use involving groups different from those that actually used the screen-sharing awareness tool during the evaluation. Finally, I conclude this thesis in Chapter 7 by revisiting the research questions described earlier in this chapter and then discussing opportunities for future work.

Chapter 2. Related Work

In this chapter, I present the background behind this thesis and give a brief overview of related work. First, I describe the importance of informal awareness and casual interaction in everyday coordination and work. I then discuss why a lack of informal awareness and casual interaction is a problem for distributed groups. Next, I review the existing literature on informal awareness and casual interaction tools that are designed to address this 'distance' problem. As part of this review, I highlight how most of these systems emphasize awareness of people, paying little attention to awareness of people's work artifacts. Finally, I give an overview of screen sharing – a common way for distributed groups to collaborate over shared artifacts in real-time – as screen sharing forms the basis of my approach for providing distributed groups with artifact awareness, a component of informal awareness.

2.1 Everyday Coordination and Work

Casual interaction, the spontaneous and informal meetings that occur during the day, is important in everyday coordination and work. It is made possible by informal awareness, the naturally gained understanding of who is around, what they are doing, and whether or not they are available for conversation or collaboration (Kraut et al., 1988). In this section, I summarize the characteristics of informal awareness and casual interaction, and explain how artifact awareness is an important component of informal awareness. I then discuss the problems that arise when collaborators become separated by distance.

2.1.1 Casual Interaction

Casual interactions happen frequently throughout the day between co-located people who work in a shared environment. For example, two people might run into each other in the hallway and stop to chat, or a person walking past another's desk may remember something he or she wanted to ask the other. Whittaker et al. (1994) observed that office workers spent 31% ("a large proportion") of their work time in informal, unplanned conversations. Many of these were brief, lasting a few minutes or less, and most were dyadic. These casual interactions, while seemingly mundane, are critical to everyday coordination and work. In particular, Kraut et al. (1988) and Whittaker et al. (1994) found that engaging in casual interaction is important for:

- ***Coordinating and tracking progress:*** Collaborators working together on a project can check each other's progress and coordinate their activities. For example, Kraut et al. found that project management among researchers was extremely informal, with coordination and progress updates occurring in informal conversations along a hallway or in a lunchroom just as often as in formal, scheduled meetings.
- ***Exchanging knowledge:*** During casual conversations, people discuss various topics, such as what they are currently working on or something that has caught their interest recently. Through this exchange of knowledge, new collaborations can also form.
- ***Seeking information:*** Someone needing help may walk around until they find a colleague available who can assist them, using casual conversation with people they come across to determine who might have that knowledge.
- ***Building relationships:*** Many casual interactions involve exchanging greetings or pleasantries. Through these exchanges, people can come to know one another better. Also, through frequent casual interactions, people may come to like each other more, and consequently collaborate more together (Kraut et al., 1988).

2.1.2 Informal Awareness

Informal awareness makes casual interaction possible; knowing who is around, what they are doing, and whether or not others are available for conversation or collaboration helps people find opportunities for and decide whether to initiate casual interaction with others.

Informal awareness often comes naturally when people are in a co-located environment because people naturally gather visual and auditory cues about other people's presence and activities. For example, subtle cues such as the sound of a door closing or seeing someone walk by wearing their coat and hat can provide information about their current or future availability.

2.1.3 Artifact Awareness

An important component of informal awareness is *artifact awareness* – one person's knowledge of the artifacts and tools that other people are working with. For office workers, artifacts include the documents and drawings (both physical and digital) that people work on over the course of a day, the secondary materials that support their tasks, and the tools they use to carry out their work. Being aware of these artifacts is valuable for a variety of reasons:

- ***Monitoring and coordinating:*** Collaborators who are responsible for different aspects of a joint task can monitor each other's progress and coordinate their activities.
- ***Triggering interest:*** Seeing another person's activity, even if it is not part of a joint task, can trigger interest in that activity. For example, Greenberg (1999) presented situations where people initiated interactions when a person's activity became publicly visible (such as joining a child's videogame when it was visible on a television set).
- ***Determining availability:*** Knowledge of artifacts is yet another source of contextual information that helps people determine how busy others are and when they can be interrupted (Kraut et al., 1988; Whittaker et al., 1994).
- ***Creating serendipitous opportunities:*** Artifact information creates opportunities for people to engage in artifact-oriented conversations, and to move into collaboration over the artifact. For example, Whittaker et al. (1994) found that over half of all casual interactions in an office involved some form of document sharing, where

documents were mostly used as a cue or conversational prop. Similarly, Nardi (1993) found that people opportunistically collaborate over spreadsheets, CAD systems, and other documents over the course of a day.

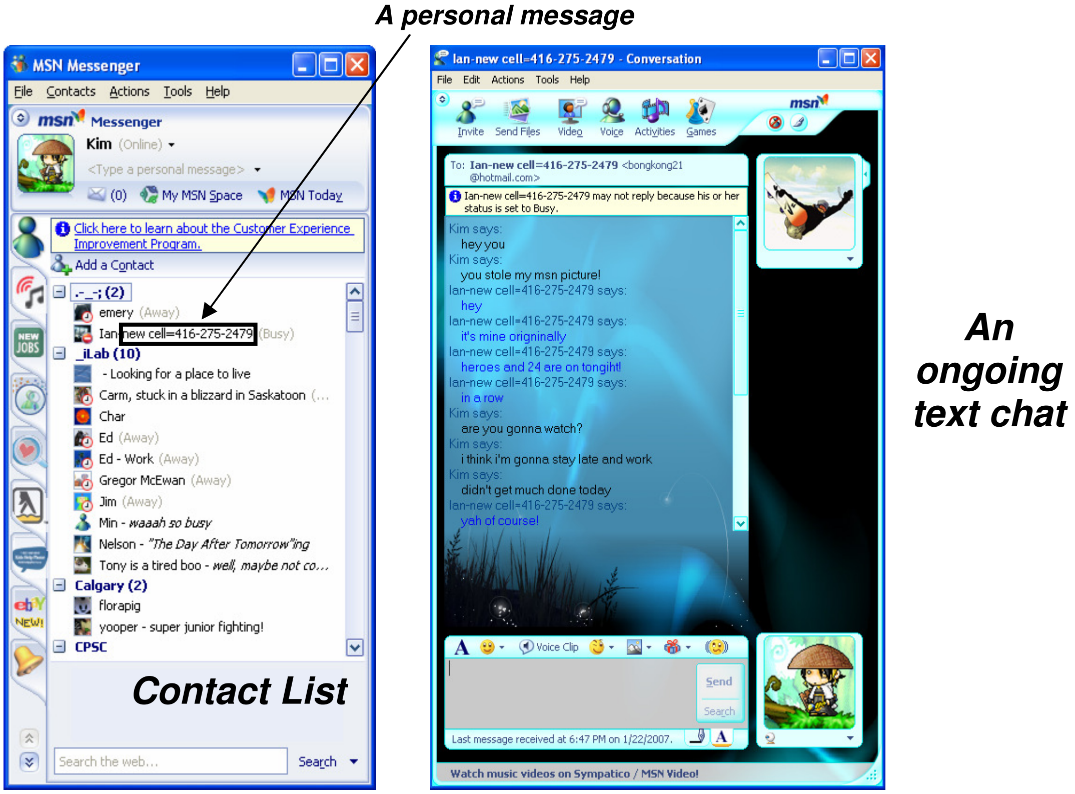
2.1.4 The Distance Problem

The difficulty for distributed groups, where members are working in different locations, is that they lack awareness cues. Consequently, they must put a relatively large amount of effort into explicitly coordinating interaction. This effort is a problem, because many opportunistic interactions would not occur if they had to be planned in advance (Kraut et al., 1988), suggesting that distributed groups are missing out on valuable opportunities for collaboration. Distributed groups have been becoming more aware of this ‘distance problem’ in recent years, however; this growing awareness partially explains the increase in number and success of low-effort awareness servers and casual interaction systems. These systems are described in more detail in the next section.

Yet, despite the recent popularity of awareness servers and casual interaction systems, distributed groups still lack the easy awareness of others’ artifacts that is normally found in a co-located shared environment. Whittaker et al. (1994) found in their study that a little over half of all casual interactions in an office involved document sharing, which strongly suggests that distributed groups would benefit from being able to share artifacts easily. Yet, most systems let people share artifacts (if at all) only after interaction has already begun.

2.2 Informal Awareness and Casual Interaction Systems

A number of ‘groupware’ tools have been developed to support informal awareness and casual interaction within small communities of distributed collaborators. These tools purportedly help distributed groups overcome the disadvantages of being distance-separated by providing them with awareness cues and opportunities for informal communication not normally available in a distributed environment. Much of the research on providing informal awareness and casual interaction to distributed groups has focused


A personal message
Contact List
An ongoing text chat
MSN Messenger
File Contacts Actions Tools Help
msn Messenger
Kim (Online)
<Type a personal message>
(0) My MSN Space MSN Today
Add a Contact
Search the web...
Search
Ian-new cell=416-275-2479 - Conversation
File Edit Actions Tools Help
Invite Send Files Video Voice Activities Games
Voice Clip
Send
Search
Watch music videos on Sympatico / MSN Video!

Click here to learn about the Customer Experience Improvement Program.

.-_-; (2)
emery (Away)
Ian-new cell=416-275-2479 (Busy)
_iLab (10)
- Looking for a place to live
Carm, stuck in a blizzard in Saskatoon (...
Char
Ed (Away)
Ed - Work (Away)
Gregor McEwan (Away)
Jim (Away)
Min - *waaah so busy*
Nelson - *"The Day After Tomorrow"ing*
Tony is a tired boo - *well, maybe not co...*
Calgary (2)
florapig
yooper - super junior fighting!
CPSC

To: Ian-new cell=416-275-2479 <bongkong21@hotmail.com>

Ian-new cell=416-275-2479 may not reply because his or her status is set to Busy.

Kim says:
hey you
Kim says:
you stole my msn picture!
Ian-new cell=416-275-2479 says:
hey
Ian-new cell=416-275-2479 says:
it's mine origninally
Ian-new cell=416-275-2479 says:
heroes and 24 are on tongiht!
Ian-new cell=416-275-2479 says:
in a row
Kim says:
are you gonna watch?
Kim says:
i think i'm gonna stay late and work
Kim says:
didn't get much done today
Ian-new cell=416-275-2479 says:
yah of course!

Last message received at 6:47 PM on 1/22/2007.

Figure 2.1 – MSN Messenger.

on instant messaging, chat rooms, and media spaces. In this section, I give an overview of prior research done on these types of systems.

2.2.1 Instant Messaging Systems and Chat Rooms

Instant messaging (IM) systems primarily support real-time text chat across the Internet. They are extremely popular both at home and in the workplace, and are being used by millions of people worldwide for social and work purposes (Isaacs, Walendowski, Whittaker et al., 2002). MSN Messenger (Figure 2.1), ICQ, and AOL Instant Messenger are examples of some of the more widely-used systems.

IM systems consist of contact lists (Figure 2.1, left), which contain information about the people that users can communicate with over IM. Typically, contact lists provide *presence* indicators of the people currently connected to the network and *status* indicators showing whether they are busy or available for conversation. The status indicators can be set deliberately by users or be automatically set by the system based on how long input devices have been idle. Status indicators can be unreliable, for example showing someone as online and available when they have recently left their office, or showing someone as absent when they are in fact present but not using their computer (e.g., they are reading at their desk). Thus, someone who does not respond immediately to a message is not necessarily seen as rude by the sender, since senders do not know for certain whether recipients are at their computer. This *plausible deniability* allows message recipients to respond at their convenience (Nardi et al., 2000). Some contact lists also provide additional information for each contact on the list, such as a personal message set by that person (Figure 2.1, top). This information, as well as the presence information, is valuable to people for creating and maintaining a sense of social connection to those on their lists (Nardi et al., 2000; Smale and Greenberg, 2005). Without even having to interact with them, people are able to get a sense of others, such as how they are feeling, what they are doing, or where they are.

In order to begin a conversation with a person on their contact list, one simply double-clicks on that contact's name. This opens a separate chat window where text messages can be composed and viewed (Figure 2.1, right). Although most conversations are dyadic, it is not uncommon for users to have multiple concurrent dyadic conversations with different contacts in separate chat windows (Cameron and Webster, 2005; Grinter and Palen, 2002). Additional users can be added to a conversation to form a group chat. Close friends or co-workers may have overlapping contacts in their list, but generally, every contact list consists of different social and work contacts. Consequently, IM systems do not broadcast conversations to those not involved in them.

While the awareness information provided by IM is very basic, it succeeds because it lets people easily establish communication with one another at opportune times. As a result of this, people use IM for a number of purposes, including: coordinating work tasks, asking quick questions, coordinating impromptu social meetings, and keeping in touch with friends and family (Isaacs, Walendowski, Whittaker et al., 2002). This casual interaction is similar to that which occurs in a co-located environment. However, unless a contact on a list explicitly sets a personal message for others to view, there is no awareness of what that contact's current activities are.

Until recently, IM conversations were restricted to text chat. Newer systems offer richer communication channels such as internet telephony (VoIP) and video, groupware applications such as shared drawings, games and shared screens, and information exchange via file transfer. In some systems, it is also possible to add audio or a live video feed from a webcam to a conversation. These extra channels are not normally used for awareness, but instead to augment a previously-initiated conversation or activity.

Some research IM systems improve upon the basic awareness information available in mainstream IM systems by providing additional information and cues. ConNexus (Tang et al., 2001) augments IM with functionality such as calendar sharing and awareness information such as whether the user is currently engaged in any activity, for example a chat with another user. Hubbub (Isaacs, Walendowski, and Ranganthan, 2002) uses sound cues in addition to visual cues to provide background awareness information. For example, each user chooses a short segment from a song as their "auditory name" that plays whenever they send a message or become active after being idle or offline, so other users can easily identify messages from them, or changes in their status. While both ConNexus and Hubbub were found to support informal awareness and casual interaction on a person-to-person basis, there was little or no support for spontaneous group interaction in either system. There was also little support for awareness of artifacts or individual work.

In contrast to IM, which primarily supports casual interaction between personal contacts in mostly dyadic conversations, *chat rooms* are public places where all can see and post messages. One example is Rear View Mirror (Handel and Herbsleb, 2002), a mix of an IM system and a chat room, which emphasises group-oriented functionality. Rather than having contact lists defined independently by each user, it uses groups whose membership is determined by a group administrator. The contents of group chats are available to all members of the group, whether or not they are participating. This focus on groups is also seen in media spaces, discussed in the next section. During evaluation, the group chats were found less intrusive than IM, since the messages were usually directed toward a group or other individuals, so users didn't feel compelled to look at new messages immediately if they were already occupied. While users were able to get some awareness of others' work when it was discussed during group chats, Rear View Mirror still primarily provided awareness of people, and not of their artifacts.

Other examples of chat rooms include: text-based chat rooms such as Internet Relay Chat (IRC), where conversations often happen between people whose real world identities are unknown to each other; graphical chat rooms such as Comic Chat (Kurlander et al., 1996), where people can create avatars to represent themselves while they meet and participate in conversations with others; and virtual worlds, where people can view and manipulate visual artifacts that comprise that virtual world (Bartle, 2004). While there can be strong awareness of the group and/or virtual shared artifacts within chat rooms, it is uncommon for people to use them to have work-related conversations or to share artifacts with their co-workers.

In summary, IM systems and chat rooms enable lightweight communication between people, typically through text chat. Opportunities for these interactions are created through very basic awareness of others' online presence. While this awareness helps make casual interaction easy, the awareness provided is typically only of people, not of people's artifacts or individual work. In the next section, I discuss media spaces, which provide richer awareness of people and their environments.

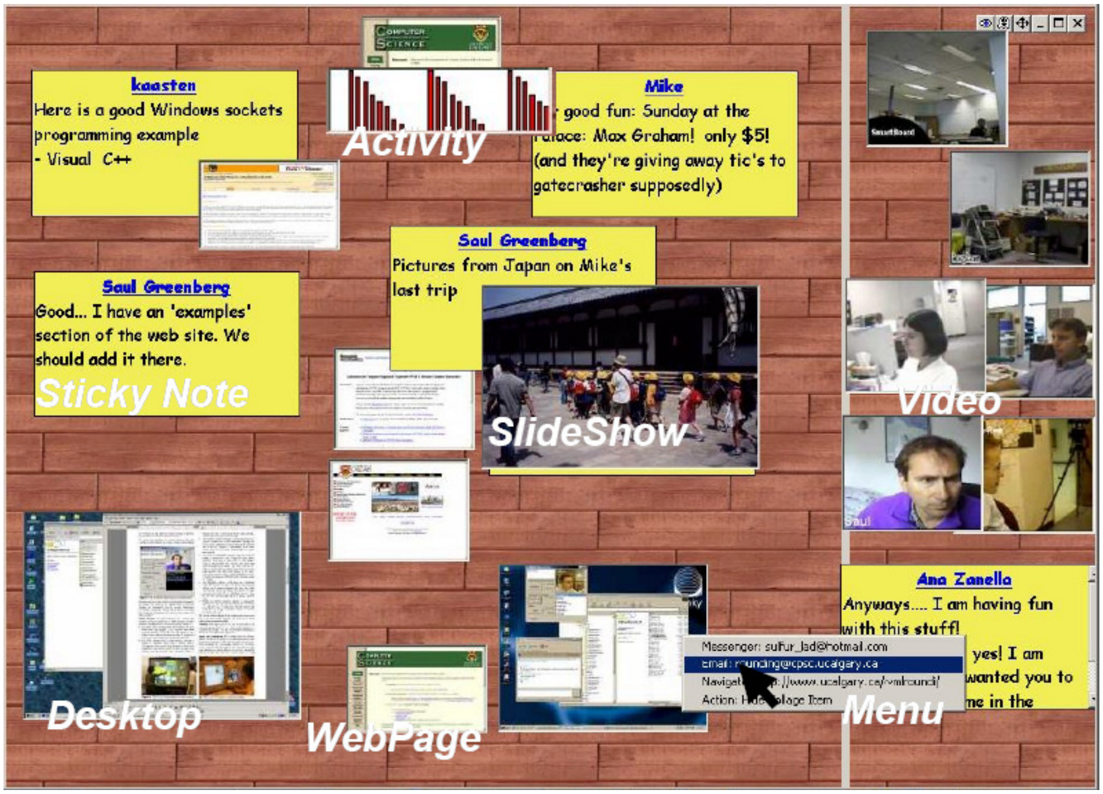
2.2.2 Media Spaces

Media spaces link offices and public spaces through networks of audio and video to provide rich awareness of people and their immediate surroundings (Bly et al., 1993). The resulting collection of "always-on" videos can be shown on a personal computer, on a dedicated television monitor (for earlier analogue systems), or even on a video wall placed in a common area. By seeing others through the media space, people get a sense of others' presence and availability, their social interactions, and sometimes their activities.

One of the earlier media spaces was CAVECAT (Mantei et al., 1991), which connected individual offices and public spaces using full audio and video to support distributed meetings and collaborative work. Each workstation that connected to CAVECAT required a video camera, microphone, and television monitor. The television monitor could display up to four video images at a time. CAVECAT was commonly used for video conferencing between groups of people, as well as for casual interactions such as introducing visitors to others without having to actually walk over to others' offices. However, there were technical problems such as delays in establishing connections and poor audio filtering that made usability of the system problematic. There was also little consideration for privacy concerns.

Cruiser (Root, 1988; Fish et al., 1992) was another early media space that provided full audio and video. In addition to supporting awareness and interaction between members of a distributed group, Cruiser was also designed to encourage spontaneous conversation through chance encounters: the system could initiate calls or video-only connections between users at random times, simulating situations where people run into each other in a hallway or glance through office doors while walking past. Users could then choose whether or not to engage in interaction. In practice, these system-initiated random encounters were found by users to be very disconcerting, as there was no transition stage before users were "suddenly confronted with another compelled conversation" (Fish et al., 1992). Another problem, reported by 90% of the participants in Fish et al.'s study, was that Cruiser did not allow any way of sharing work artifacts. While Cruiser enabled casual conversation, it did not allow transitions into work-related talk or focused collaboration around work artifacts.

More recent media spaces 'relax' the notion of video by instead providing occasionally updated snapshots of a group of people, perhaps mixed with other media. Portholes (Dourish and Bly, 1992) was the first example of such a media space; it uses periodic snapshots of multiple offices and common areas to support collective awareness in a distributed group. Snapshots are arranged as a grid of "windows" into someone else's space.



kaasten
Here is a good Windows sockets programming example
- Visual C++

Mike
good fun: Sunday at the
alace: Max Graham! only $5!
(and they're giving away tic's to gatecrasher supposedly)

Saul Greenberg
Pictures from Japan on Mike's last trip

Saul Greenberg
Good... I have an 'examples' section of the web site. We should add it there.

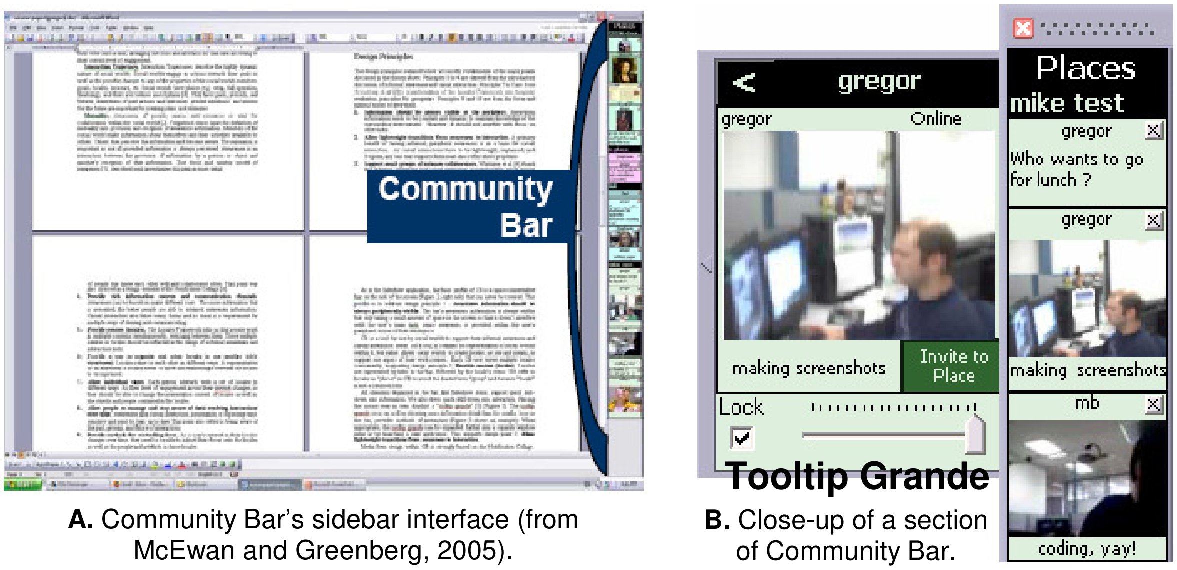
Ana Zanella
Anyways.... I am having fun with this stuff!
yes! I am
wanted you to
me in the

Activity
Sticky Note
SlideShow
Video
Desktop
WebPage
Menu
Messenger: sulfur_lad@hotmail.com
Email: rounding@cpsc.ucalgary.ca
Navigate: p://www.ucalgary.ca/~vmround/
Action: Hide Collage Item
SmartBoard
Saul

Figure 2.2 – Notification Collage (from Greenberg and Rounding, 2001).

Another example of a snapshot-based media space that also supports casual interaction is the Notification Collage (Greenberg and Rounding, 2001). Members of a distributed group can post media elements that include not only live video snapshots but also editable sticky notes, photo slideshows, and web page thumbnails onto a real-time surface that all members can see (Figure 2.2). It mimics a public bulletin board by placing posted elements randomly onto the surface.

Through informal evaluations, both Portholes and the Notification Collage were found to be able to improve awareness of others' presence and activities, as well as improve the "connection" or "sense of community" between remote users (Dourish and Bly, 1992; Greenberg and Rounding, 2001). Because of the variety of media elements available, the Notification Collage in particular provided a rich experience; in addition to providing informal awareness of others, the Notification Collage also allowed members to communicate spontaneously with each other and to share artifacts and information about their personal lives. However, one of the most common complaints by users of both



Who wants to go for lunch ?

A. Community Bar's sidebar interface (from McEwan and Greenberg, 2005).

B. Close-up of a section of Community Bar.

Figure 2.3 – Community Bar.

media spaces was that they took up a lot of screen space. A later version of the Notification Collage allowed transparency so that users could still work through it, though this was only a partially successful solution. In practice, the expectation was that people would have two monitors, one containing the Notification Collage.

Community Bar (McEwan and Greenberg, 2005) is a direct descendent of the Notification Collage. It is not only space-compact (addressing the display room issue), but is also designed around sociological theory. It consists of an always-visible vertical bar that resides on one side of the screen (Figure 2.3a), and displays various media items such as presence indicators (Figure 2.3b, middle and bottom), web page thumbnails, and text chat (Figure 2.3b, top). Like the Notification Collage, items posted to Community Bar are visible to all members of the group, enabling group awareness and communication. Because there is limited space in the sidebar to display information, Community Bar enables people to transition from awareness to exploration to interaction with information posted to it. For example, if a user wishes to find out more about a particular media item in the sidebar, they can place their mouse over it and a *tooltip grande* containing additional information and interaction opportunities will appear (Figure 2.3b). A more detailed description of Community Bar appears in Section 3.1, as the screen-sharing awareness tool described in this thesis is prototyped as a Community

Bar media item in order to exploit Community Bar's existing support for informal awareness and casual interaction.

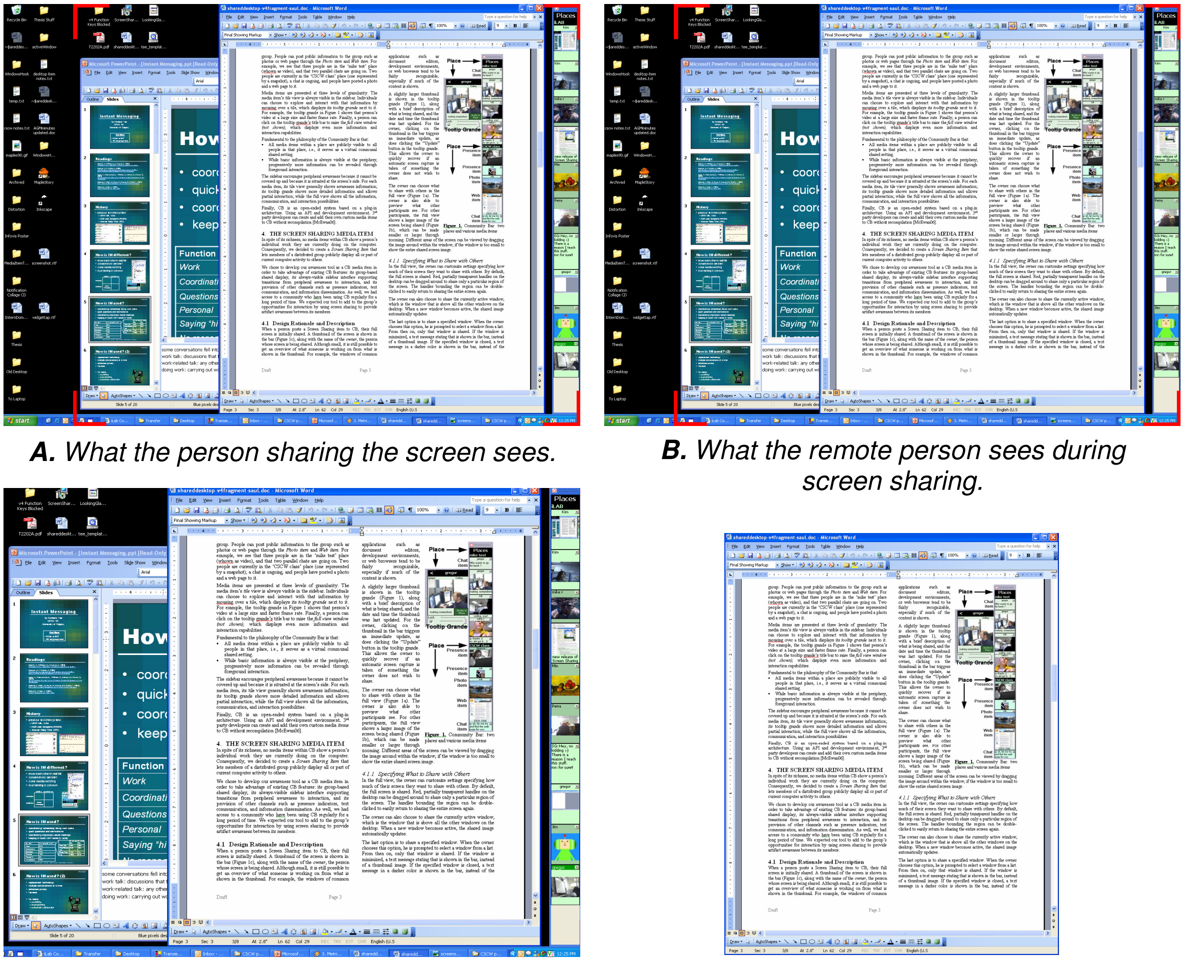
In summary, studies of media spaces have shown that they are somewhat successful at building and maintaining the sense of "connection" or "community" between people at different locations. However, they have mainly focused on doing so through video and chat. While some media spaces also allow people to share web sites and photos, it is uncommon for them to enable people to easily share dynamic, work-related artifacts such as reports, spreadsheets, code, or other documents. Instead, artifacts are usually explicitly posted for disseminating information of interest to the group, rather than to provide awareness of ongoing activities.

2.3 Screen Sharing

One way of sharing work-related artifacts that has been used in real-time distributed collaboration is to share one's screen, window, or window fragment with another person (Figure 2.4). This emulates over-the-shoulder sharing of a computer: one person can show others what they are working on, and each can take turns interacting with the system. As screen sharing forms the basis of my approach for providing distributed groups with artifact awareness, in this section, I give a brief history and overview of screen sharing.

2.3.1 A Brief History

Screen sharing was first demonstrated as part of the NLS same-room conferencing system in 1968 (Engelbart and English, 1968). One person controlled what was being presented, but others could see and point at things from their own machines by using telepointers. Engelbart also showed how two distance-separated people could see a shared display, talk over an audio channel, and see each other through quarter-frame video (Engelbart and English, 1968). Screen sharing later moved into commercial practice, with early systems appearing in the mid 1980's, e.g., Farallon Timbuktu (WOS Data Systems, 1987) and VNC (Richardson et al., 1998). In the late 1980's and early



***A.** What the person sharing the screen sees.*

***B.** What the remote person sees during screen sharing.*

***C.** What the remote person sees during region sharing, where person (a) is sharing the region specified by the red handles.*

***D.** What the remote person sees during window sharing, where person (a) is sharing the* Microsoft Word *window.*

Figure 2.4 – Screen, window, and region sharing.

1990's, researchers revisited screen sharing, including architectural issues and platform-specific solutions (see Section 2.3.3).

2.3.2 Focused Collaboration vs. Awareness

Screen sharing systems were first used to augment face-to-face interactions, but soon after, they were also used for (single-user) remote access to a computer from another location and for desktop conferencing, in which non-computer-supported audio/video teleconferencing technologies were integrated with desktop computers so that individuals could meet, collaborate, and work together from their offices. Now, other common uses

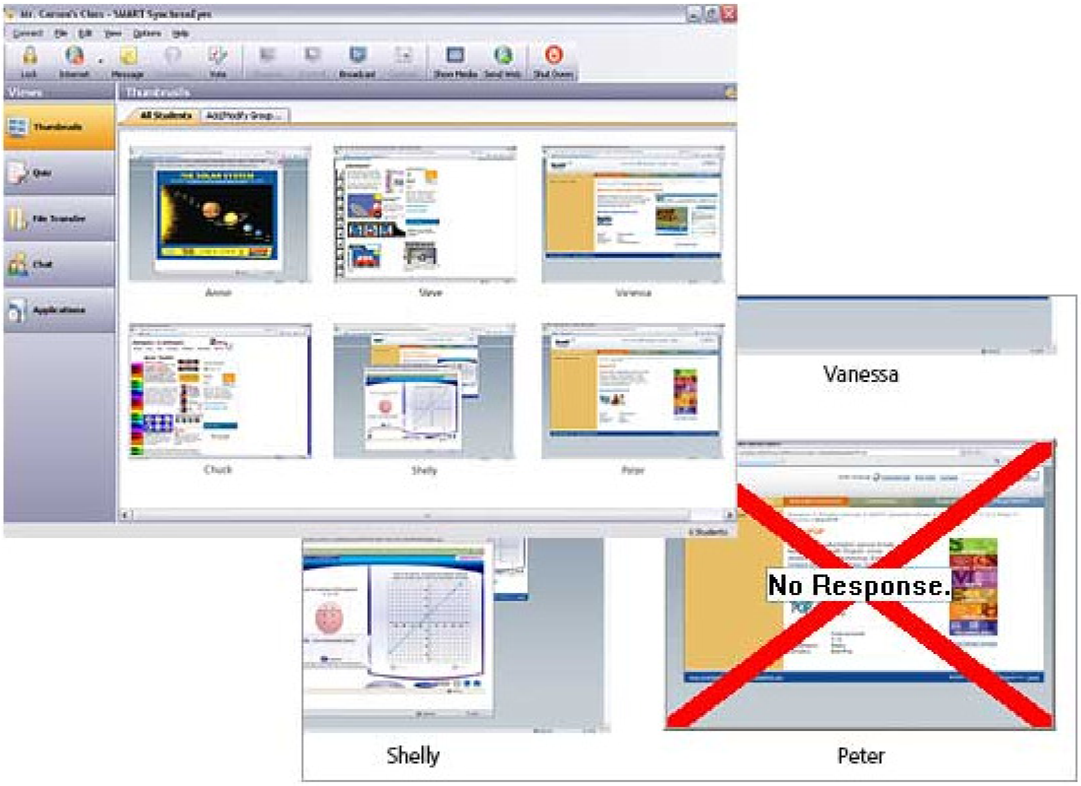


Figure 2.5 – SynchronEyes [from www.smarttech.com].

of screen sharing include application sharing (desktop conferencing without requiring the use of audio/video – telephones typically support the talk) and remote assistance (a system administrator or an expert user can remotely control another's computer to assist them in performing certain tasks).

Yet, despite almost twenty years of use, little work has been done investigating the use of screen sharing for providing awareness. Instead, current screen sharing systems are designed primarily for focused interaction, rather than artifact awareness leading to casual interaction. One exception is SynchronEyes [www.smarttech.com], a commercial system technically closest to the screen-sharing awareness tool that I prototyped. It too lets one person view others' desktops as thumbnails (Figure 2.5). However, it is designed for a quite different educational setting where a teacher monitors / controls a class of students and not as a peripheral awareness tool used by peers. To the best of my knowledge, no previous research has considered screen sharing as an awareness mechanism leading to casual interaction.

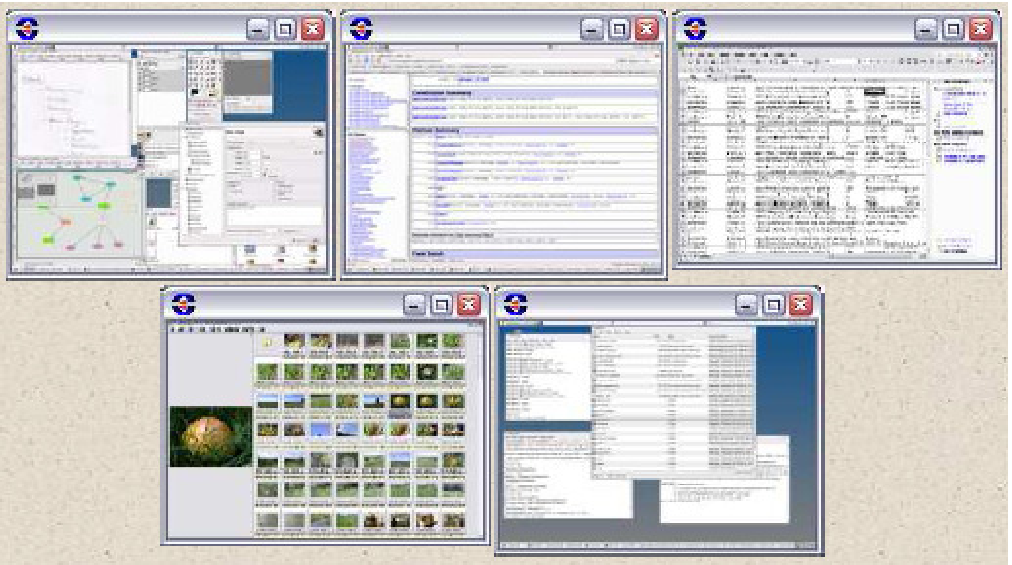


Figure 2.6 – Multi-VNC (from Gutwin et al., 2005).

Some preliminary work exploring the idea of screen sharing for awareness was first seen in the Notification Collage Desktop media item (Rounding, 2004): people could post an intermittently updating desktop thumbnail (Figure 2.2, bottom left) that others could see and select if desired to activate a ‘full’ screen sharing session, one in which remote participants are able to act upon the screen / applications being shared. The Desktop media item did the basic screen capture, but moving into full screen sharing was done by invoking the third-party external Bridgit software [www.smarttech.com] for full screen sharing. This media item was originally built to show that third-party software could be easily linked into the Notification Collage system; it did not consider issues such as privacy, for example – it simply shared the entire desktop.

Multi-VNC (Gutwin et al., 2005) later followed, as a proof-of-concept prototype in which multiple instances of a screen sharing application, UltraVNC [ultravnc. sourceforge.net], were run so that each member of a group could see the other members’ computer desktops (Figure 2.6). However, the system was limited to the functionality that UltraVNC could provide. For example, Gutwin et al. (2005) point out that commands such as scaling and zooming of the desktop image have to be given through the UltraVNC control dialogs, which is difficult and time-consuming. Also, there is no way to determine which workspace belongs to whom, or whether someone is active or away from their computer. In essence, neither Multi-VNC nor the Notification Collage Desktop media item examined in detail how to provide awareness using screen sharing; at best, they were proof-of-concept illustrations.

2.3.3 Implementation Approaches

Because this thesis exploits screen sharing methods, I give a brief discussion here of different ways to implement screen sharing. This genre of groupware is often called 'collaboration-transparent', as the underlying application or screen being shared has no knowledge that multiple people are viewing and/or using it, i.e., special software wrapping the underlying application or screen being shared is used to handle input and output between the remote participants' systems and the shared application / screen.

This special wrapper software is known as the *manager* (Greenberg, 1990); by accepting input from any participant's system as input to the shared screen and by distributing the output from the shared screen to the participants' systems, a shared workspace is created in which each participant sees the same view and each participant can interact through their local input devices with the shared machine. Besides handling input and output, the manager is also responsible for a number of other tasks, such as handling latecomers who join the shared session after it has begun, setting and changing floor control, coordinating turn-taking, and controlling meta-actions such as participants' gestures and annotations that do not affect the shared screen (Greenberg, 1990; Lauwers and Lantz, 1990).

Today, screen sharing systems typically follow a *centralized* architecture, which consists of the machine sharing its application (or screen), the remote participants' systems, and a manager that handles the input and output between them (Figure 2.7). When participants interact with the shared application, all the input is redirected to the manager. The manager then consolidates or filters the input in some way before forwarding it to the shared application. Once the shared application has produced some output or updated its view, it sends the output information to the manager, which then distributes it to the various participants' systems.

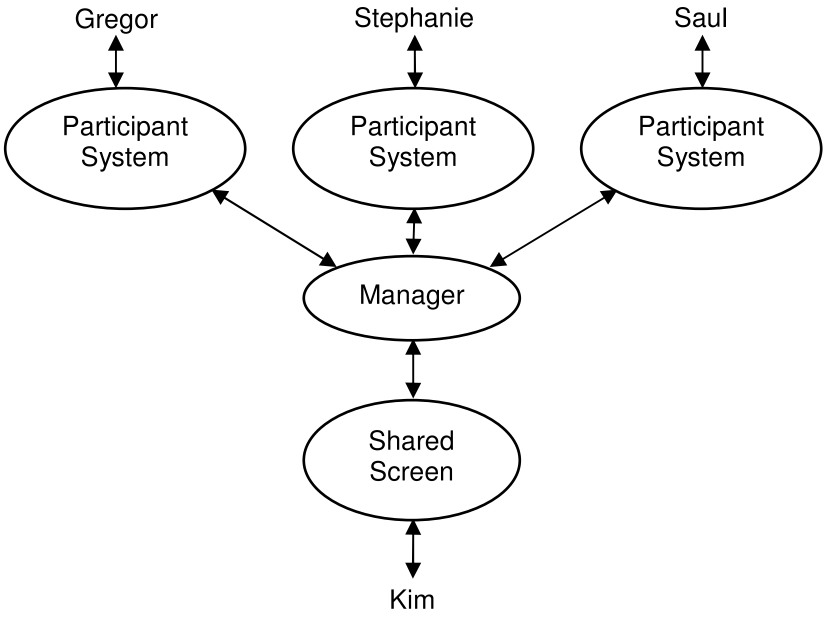


Figure 2.7 – Components of a centralized screen sharing system (modified from Lauwers and Lantz, 1990).

In contrast, in a *replicated* architecture, there are as many instances (or replicas) of the manager and each application being shared as there are participants in the session. When participants interact with their own replica of the shared application, the input is also sent to the replica's corresponding manager. That manager synchronizes with the different managers from each participant's system, which then forward the synchronized input to their own replicas of the shared application. Each replica produces its own output, which can then be sent directly to its user without having to be redirected through the manager. A screen sharing system using a replicated architecture usually provides better performance than one using a centralized architecture since each machine processes the input locally, so slower machines do not hold up the faster ones. However, the replicas in a replicated environment must be synchronized at all times, which can be difficult to achieve, so most systems tend to be centralized (Lauwers and Lantz, 1990).

In a centralized screen sharing system, one of the primary concerns is what output the shared machine's manager should transmit to the remote participants' systems over the network. Typically, the shared machine's manager sends either graphics commands (that instruct the participants' systems on what to draw on their screens) or raw images (that are simply displayed as is) to participating systems. One example of a system that sends graphics commands as output is Shared X (Garfinkel et al., 1989). Its manager sends graphics commands using the “X Protocol” to participating systems, telling them what to draw on their displays. This approach can be efficient because sending commands across the network does not require much bandwidth. However, there are also a number of potential problems with this approach: a participating system may not understand the graphics command, e.g., Macintosh commands received by a Windows machine; a participating system may not be able to perform a command, e.g., fonts on one machine may not be available on another or keyboard keys may be mapped differently; and different displays may have different capabilities, e.g., screens may be different sizes, or can only display black and white vs. colour (Garfinkel et al., 1989). Thus, image-based screen sharing remains the most common approach used today.

Consequently, a number of optimizations have been developed for improving the performance of systems that use image-based screen sharing. For example, the MBlink system (Sarin, 1984) sends only the pixel differences between images across the network, rather than the full image. nv (Frederick, 1994) adaptively uses various encoding schemes for compressing the pixel data to optimize for network bandwidth and server processing speed. VNC (Richardson et al., 1998) also uses various encoding schemes for pixel data to optimize for network bandwidth, server processing speed, and viewer drawing speed.

The artifact awareness tool that I built follows a centralized, image-based approach to screen sharing. This approach was easiest to implement, and as the awareness tool needed to be able to provide different-sized views of each screen being shared, an image-based approach was particularly appropriate. This approach was also ‘good enough’ in terms of performance. Specific details of the implementation are provided in Chapter 4.

2.4 Summary

In this chapter, I have described the motivation and background behind my research. Casual interaction is important in everyday work for exchanging knowledge and building relationships, and is made possible by informal awareness. Artifact awareness is an important component of informal awareness, defined as one person’s knowledge of the artifacts and tools that other people are working with. By having awareness of others' individual work and artifacts, people can easily monitor and coordinate joint work, determine whether others are available, and create serendipitous opportunities for conversation and collaboration. In a co-located setting, artifact awareness can already be difficult, particularly when the artifacts are digital; when people are separated by distance, they lose the cues that make informal awareness and artifact awareness possible, and as a result, casual interaction becomes more difficult.

Informal awareness and casual interaction tools attempt to address this distance problem by providing a means for informal awareness and casual interaction to occur in distributed groups. Some examples of these tools are IM systems, chat rooms, and media spaces, which provide varying levels of awareness and interaction. However, these systems primarily provide awareness of people, not artifacts. Even augmented media spaces such as the Notification Collage and Community Bar that provide awareness of both people and artifacts do not share awareness of people's work. The artifacts being shared in these systems are also things that people have to explicitly post.

One way of sharing work artifacts is through screen sharing, which has been around for several decades. Screen sharing systems are mainly used for providing (single-user) remote access to a computer from another location, allowing system administrators to remotely upgrade systems, and enabling remote conferencing and collaborative work in meetings. There are also groupware systems such as shared editors and webcast meeting tools that let a distributed group share artifacts. However, both these types of systems are designed for focused collaboration, rather than artifact awareness leading to casual interaction. In other words, they are typically only used *after* interaction is initiated.

In the remaining chapters of this thesis, I will draw on the literature reviewed in this chapter to investigate the use of screen sharing for providing artifact awareness. In the next chapter, I describe the screen-sharing awareness tool that I built to support artifact awareness in distributed groups. In Chapter 4, I describe how I implemented the system. In Chapter 5, I present findings from its evaluation by two groups of end-users. In Chapter 6, I report on a theoretical analysis of privacy that I did on the system.

Chapter 3. An Artifact Awareness Tool

In the preceding chapter, I discussed why informal awareness, artifact awareness, and casual interaction are important in everyday work. I described the distance problem, in which the difficulty of maintaining awareness of others and their artifacts in a distributed group makes it harder for members of the group to move into opportunistic interaction and collaboration. I also presented several types of systems designed to address this distance problem; while these systems are successful in helping distributed people communicate easily as well as feel “connected” to other group members, they currently do not support awareness of people’s individual work, instead sharing mostly presence and availability information. Though some of these systems do provide limited awareness of artifacts, users must still explicitly post a message or an image of their work in order to share it with others.

In this chapter, I present an artifact awareness tool that uses *screen sharing* to provide intimate collaborators with awareness of each others’ artifacts and individual work. It was built within the context of the Community Bar, a platform that already supports group-based informal awareness and casual interaction. Community Bar also easily enables extensions by third-party developers. However, it currently has no artifact awareness tool within it. I start this chapter with several scenarios motivating screen sharing for awareness. I then give a brief overview of Community Bar and how it works. Then, I describe the artifact awareness tool that I built and the design rationale behind it.

3.1 Motivational Scenarios

As seen in Chapter 2, much work has been done in the area of computer-supported cooperative work investigating awareness and casual interaction, so we know that awareness leading to casual interaction can have a number of benefits. By using screen sharing to provide artifact awareness between intimate collaborators (who primarily work on their computers), it is possible for group members to see what others are working on and looking at. This potentially lets them share knowledge and monitor progress, create opportunities for moving into direct interaction, and as a side effect, reinforce both work and social relationships within the group. However, negative situations might also arise, such as privacy violations or confidentiality breaches. In this section, I describe several scenarios in which using screen sharing to provide artifact awareness can be beneficial or harmful to the person sharing their screen and/or viewing another's screen. These scenarios involve two people, 'Mitch' and 'Lauren', who work together and share their computer desktops with each other. As will be illustrated in later sections, each sees a miniature overview of the other's desktop and can raise larger views if desired.

3.1.1 Beneficial Scenarios

The following scenarios are examples of situations in which screen sharing for awareness could facilitate someone's work or lead to the discovery of common interests.

Scenario 1: Monitoring and coordinating

Screen sharing for awareness can help people coordinate their work and keep track of each other's progress. For example, suppose that Mitch and Lauren are working on a paper together. In her overview of his desktop, she can see that he is working on the related work section, so she won't make any changes to that part of the paper until he has finished. He can see in his overview of her desktop that she's currently working on the methodology section and that she has added an extra paragraph about participant demographics, so he knows that he won't have to remind her to add that information anymore. By being able to see what sections the other is working on, Mitch and Lauren can avoid complicated document merges and be reassured that progress is happening with the paper.

Scenario 2: Offering or asking for assistance

Screen sharing for awareness can help people find information that they were looking for more quickly. For example, suppose that in Mitch's overview of Lauren's desktop, he notices that she has been searching for a C# widget that enables users to pick colours within her application. Recalling that he had previously implemented one, he offers to find the source code of it for her. Alternatively, suppose that Lauren remembered seeing Mitch implement one a few weeks ago in her overview of his desktop. She could then ask him if he would be able to help her out with her code now. When Mitch can see what Lauren is searching for, he is able to offer her assistance if he feels that he can help her and that he has the time to do so. When Lauren can see what Mitch and others are working on, she is able to direct her questions to the person most likely able to help her.

Scenario 3: Determining availability

Screen sharing for awareness can enable people to make a better estimate about when others can be interrupted. For example, suppose that it is past lunch time, but everyone is still at their computers. Lauren notices in her overview of Mitch's desktop that he is now checking email instead of working on his presentation, so she figures that it is okay to interrupt him and see if he wants to get something to eat. By being able to see what Mitch is working on, Lauren is able to make a better estimate about his availability.

Scenario 4: Creating serendipitous opportunities

Screen sharing for awareness can allow people to exchange information and knowledge when they might not have normally done so. For example, suppose that Lauren notices in her overview of Mitch's desktop that he is reading a paper that she just read last week. She is curious to know what he thinks of it, so when she sees him near the end, she initiates discussion with him about it. If Lauren hadn't been able to see what Mitch was reading, they may not have had that interesting conversation.

Scenario 5: Discovering common interest

Screen sharing for awareness can help people discover common ground, potentially strengthening their social bonds. For example, suppose that Mitch notices Lauren reading an article online about a video game. This video game is one of his favourites, but he never realized before that Lauren was also interested in the game. Now that he knows, it gives them something more to discuss during lunch.

3.1.2 Harmful Scenarios

The following scenarios are examples of situations in which screen sharing for awareness could violate someone's privacy or give others a negative impression of the person sharing their screen.

Scenario 6: Displaying sensitive information

Screen sharing for awareness can make it easier for people to inadvertently share sensitive information. For example, suppose that Mitch is analysing data from a user study where participants were promised that the collected information would be kept private and would only be seen by the study investigators. However, the raw data that he is currently working with still has the names attached to it, and he forgets that his screen is still being shared. Lauren, looking at the overview of his desktop, sees this information, which is a violation of Mitch's contract with the participants in his study. Because Mitch is so accustomed to sharing what he is working on, he inadvertently shares sensitive information with others that he shouldn't be sharing.

Scenario 7: Making a bad impression

Screen sharing for awareness can cause people viewing someone's screen to have a negative impression of that person. For example, suppose that Mitch tends to come to work early and stay late, taking lots of breaks throughout the day to read articles or play a quick game online. Lauren, viewing his desktop, may get the (wrong) impression that he is not taking his job responsibilities seriously, since she only sees what he is doing during the time that she is at work. By seeing Mitch read articles or play games during work hours, Lauren may feel that he isn't working hard enough.

Scenario 8: Being monitored

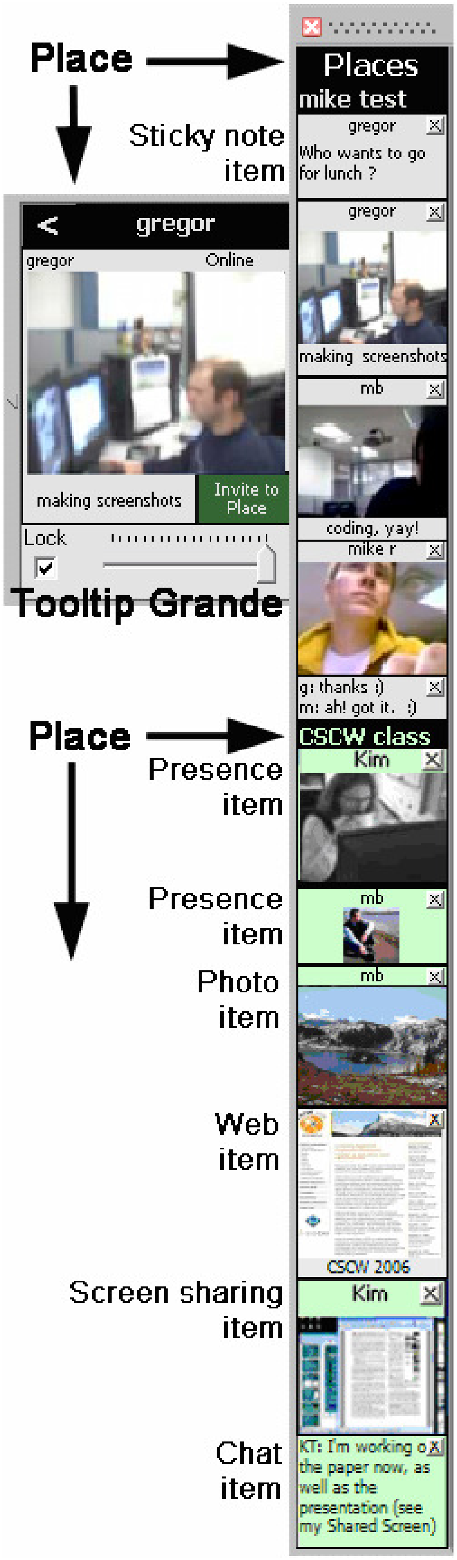
Screen sharing for awareness can be abused by (say) an office manager who insists that all employees share their desktops so that their work can be monitored. For example, suppose that Mitch and Lauren are required to share their desktops for this purpose. While they may still benefit from screen sharing for awareness, they may also feel resentful or unhappy about the implied lack of trust.

3.2 Community Bar

In the previous section, several scenarios were given to motivate the use of screen sharing for providing artifact awareness. Screen sharing forms the basis of my approach to providing groups of intimate collaborators with artifact awareness, and I discuss my approach in more detail later in this chapter. Before I describe my artifact awareness tool however, I first give a brief overview of Community Bar, an informal awareness and casual interaction tool that my artifact awareness tool was built within.

3.2.1 Overview

Community Bar (CB) is a media space that can be augmented with extra information (McEwan and Greenberg, 2005; McEwan, 2006). It provides group-based awareness of people as well as some of their artifacts, and supports transitions from awareness to interaction. It sits on one side of the screen as a vertical sidebar (Figure 2.3a), reserving space so that it is always visible as a peripheral display. Figure 3.1 shows a screen snapshot of CB in use. As can be seen, CB is divided into *Places;* two are shown in the figure. Each place represents a sub-group, their communication, their tools, and their information. These are visualized through a number of *media items,* all holding different information and all being publicly visible to the group.



Who wants to go for lunch ?

making screenshots

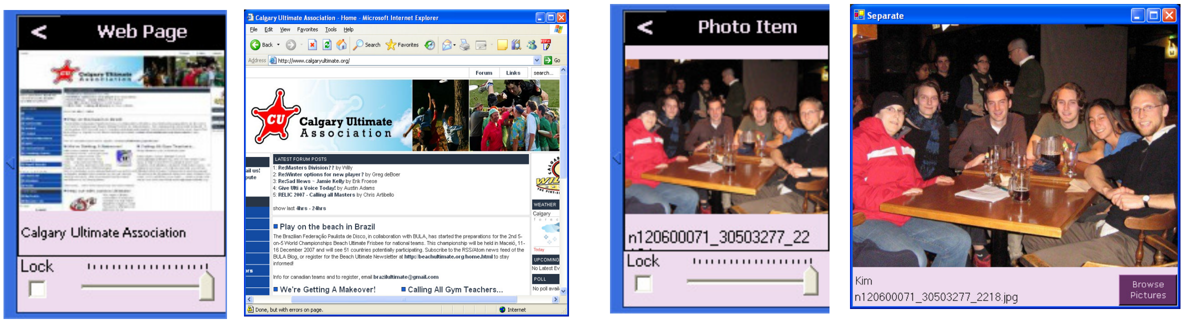
making screenshots

coding, yay!

g: thanks :)
m: ah! got it. :)

KT: I'm working on the paper now, as well as the presentation (see my Shared Screen)

Figure 3.1 – Community Bar.



A. Web item.

B. Photo item.

Figure 3.2 – Sharing artifacts in Community Bar.

As illustrated in the figure, the *Presence item* represents a person as live video, their image, or their name. Through the Presence item, others can get an idea of who is around, who is away from their desk, and who is busy. The presence awareness that people receive from this item is particularly rich when video is broadcast from desktop webcams. In fact, Romero et al. (2007) found that people's primary reason for using CB was to have this rich awareness of others' presence available to them.

Chat items hold multi-person public conversations, similar to a chat room. *Sticky Notes* contain one person's text posting to the group, much like a physical note might. People use these items to broadcast information, ask questions, or move into conversation with others. The public nature of conversations on CB makes *lurking* – where people see or overhear conversations but do not participate – possible. It also enables others to easily and serendipitously join an existing conversation.

Public information such as web pages or photos can be posted to the group through the *Web item* and *Photo item* (Figure 3.2). Through these items, CB allows people to share some of their artifacts. Though sharing websites and photos on CB takes little effort, users must still explicitly post each individual artifact that they would like to share. Also, because the Web item and the Photo item each only show one artifact at a time, the bar begins to get cluttered when people share multiple artifacts. CB currently lacks support for artifact awareness and artifact-centered serendipitous interactions; the artifact awareness tool I will introduce later in this chapter, the *Screen Sharing item*, is designed to provide support for artifact awareness and opportunistic interaction within CB.

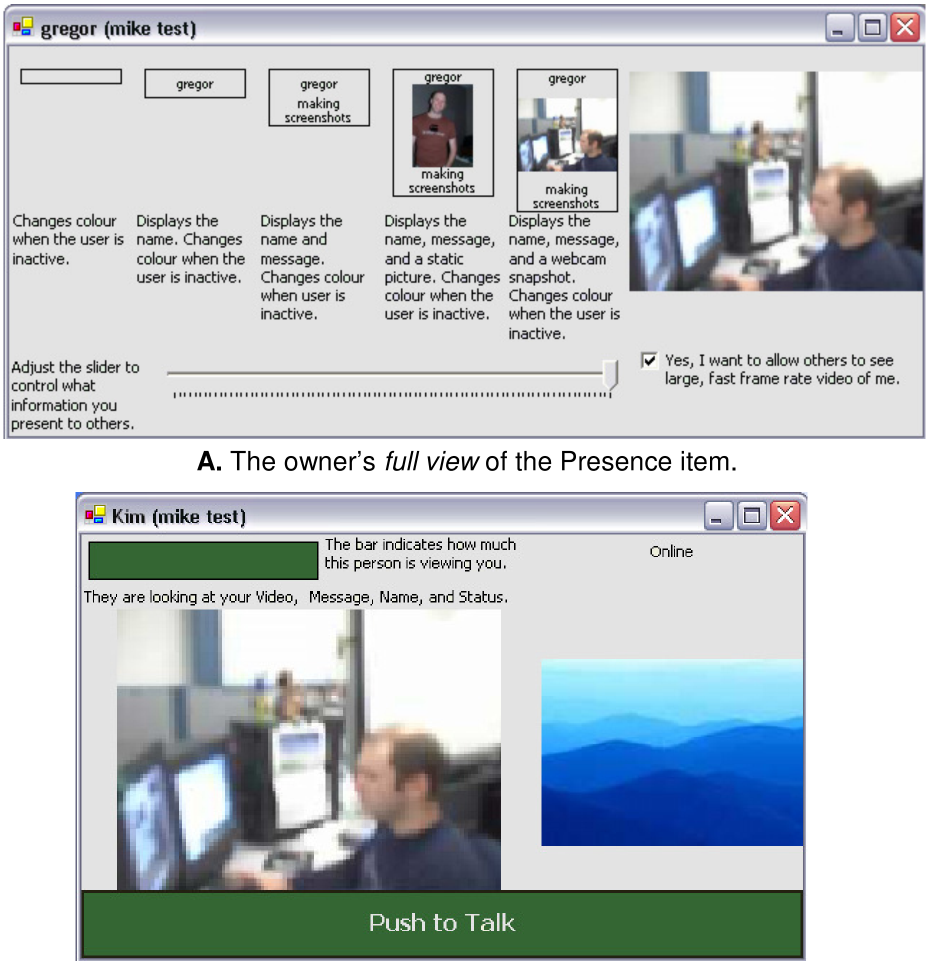
An example of CB in use is illustrated in Figure 3.1. There are three people in the ‘mike test’ place (shown as video), and two parallel chats are going on. In the ‘CSCW class’ place, there are two people (one represented by a snapshot), a chat is ongoing, and people have posted a photo and a web page to it. One person is also sharing their screen using my *Screen Sharing item*, which will be discussed later.

3.2.2 Different Levels of Awareness and Interaction

Media items are presented at three levels of granularity to provide different levels of awareness information and interaction possibilities. The media item’s *tile* view is always visible in the sidebar. If something of interest is seen in the tile, individuals can choose to explore and interact with that information in more detail by mousing over the tile, which displays its *tooltip grande* next to it. For example, the tooltip grande in Figure 3.1 shows that person’s video at a large size and faster frame rate. Finally, a person can click on the tooltip grande’s title bar to raise the *full view window* (Figure 3.3), which displays even more information and interaction capabilities. The easy transition from awareness to interaction is partly what makes CB successful (Romero et al., 2007).

Media items also have the idea of an *owner* (the person who creates the media item) and an *audience* (all others who can see the item). Depending on the media item, the view and controls available on the tile, tooltip grande, and full view may be different for the owner than for the audience. For example, the owner of a Presence item has additional controls in their full view to change what others see, e.g., a video, an image, or an icon (Figure 3.3a), whereas the audience only sees the live video stream, image, or icon of the owner (Figure 3.3b). In contrast, all people see a Chat item in exactly the same way.

Fundamental to the philosophy of CB is the idea that all the media items within a place are publicly visible to all the people in that place, i.e., it serves as a virtual communal shared setting. Also, while basic information is always visible at the periphery, progressively more information can be revealed through focused interaction. The sidebar encourages peripheral awareness because it cannot be covered up and because it is situated at the screen’s side. For each media item, its tile view generally



gregor (mike test)

gregor

gregor
making screenshots

gregor
making screenshots

gregor
making screenshots

Changes colour when the user is inactive.

Displays the name. Changes colour when the user is inactive.

Displays the name and message. Changes colour when user is inactive.

Displays the name, message, and a static picture. Changes colour when the user is inactive.

Displays the name, message, and a webcam snapshot. Changes colour when the user is inactive.

Adjust the slider to control what information you present to others.

☑ Yes, I want to allow others to see large, fast frame rate video of me.

A. The owner's *full view* of the Presence item.

Kim (mike test)

The bar indicates how much this person is viewing you.

Online

They are looking at your Video, Message, Name, and Status.

B. The audience's *full view* of the Presence item.

Figure 3.3 – *Full view* windows (modified from McEwan and Greenberg, 2006).

shows awareness information; its tooltip grande shows more detailed information and allows partial interaction; while the full view shows all the information, communication, and interaction possibilities. Notifications of new information in a media item are done through a change of background colour in the tile view.

3.2.3 Custom Media Items

While CB already includes many media items, it has been designed as an open-ended system based on a plug-in architecture. Using an API and development environment, third-party developers can create and add their own custom media items to CB without recompilation (McEwan et al., 2006). This is what was done with the artifact awareness tool, in order to take advantage of existing CB features: its group-based public display, its always-visible sidebar interface supporting transitions from peripheral awareness to interaction, and its provision of other communication and information channels such as presence indicators, text communication, and so on.

3.2.4 Community Bar Summary

As can be seen, CB provides awareness of both people and artifacts. However, the limited artifact awareness it provides is different from what I would like to offer with my artifact awareness tool, the Screen Sharing item. In spite of its richness, none of the media items within CB shows the individual work that a person is currently engaged in on their computer. While CB lets people post websites or photos of interest to the group, the artifacts that people share are typically personal in nature, rather than work-related. Also, people must individually post each artifact that they would like to share, which, though not difficult, requires some effort. CB currently lacks support for work-related artifact awareness and artifact-centered serendipitous interactions; in the next section, I describe the artifact awareness tool that I built within the context of CB to provide artifact awareness to groups of intimate collaborators.

3.3 The Screen Sharing Item

The *Screen Sharing item* artifact awareness tool lets intimate collaborators publicly display all or part of their current computer screen to others. It is intended to work within the context of other interpersonal awareness information and public conversations on CB. The following subsections describe by scenario how the Screen Sharing item works.

3.3.1 Tile View

'Kim' (initials KT) is working on a paper of interest to her group, so she decides to share her display with them. Through the CB place's context menu, she invokes the Screen Sharing item. A tile view is immediately added to that CB place, which contains a thumbnail of her entire screen labelled with her name (Figure 3.1, second tile from the bottom). At the same time, she adds a Chat item (Figure 3.1, bottom tile) saying "I'm

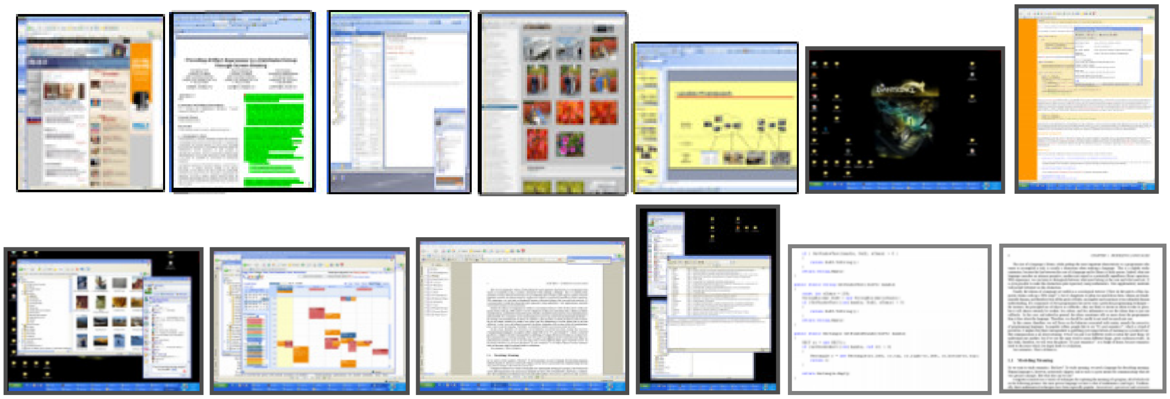


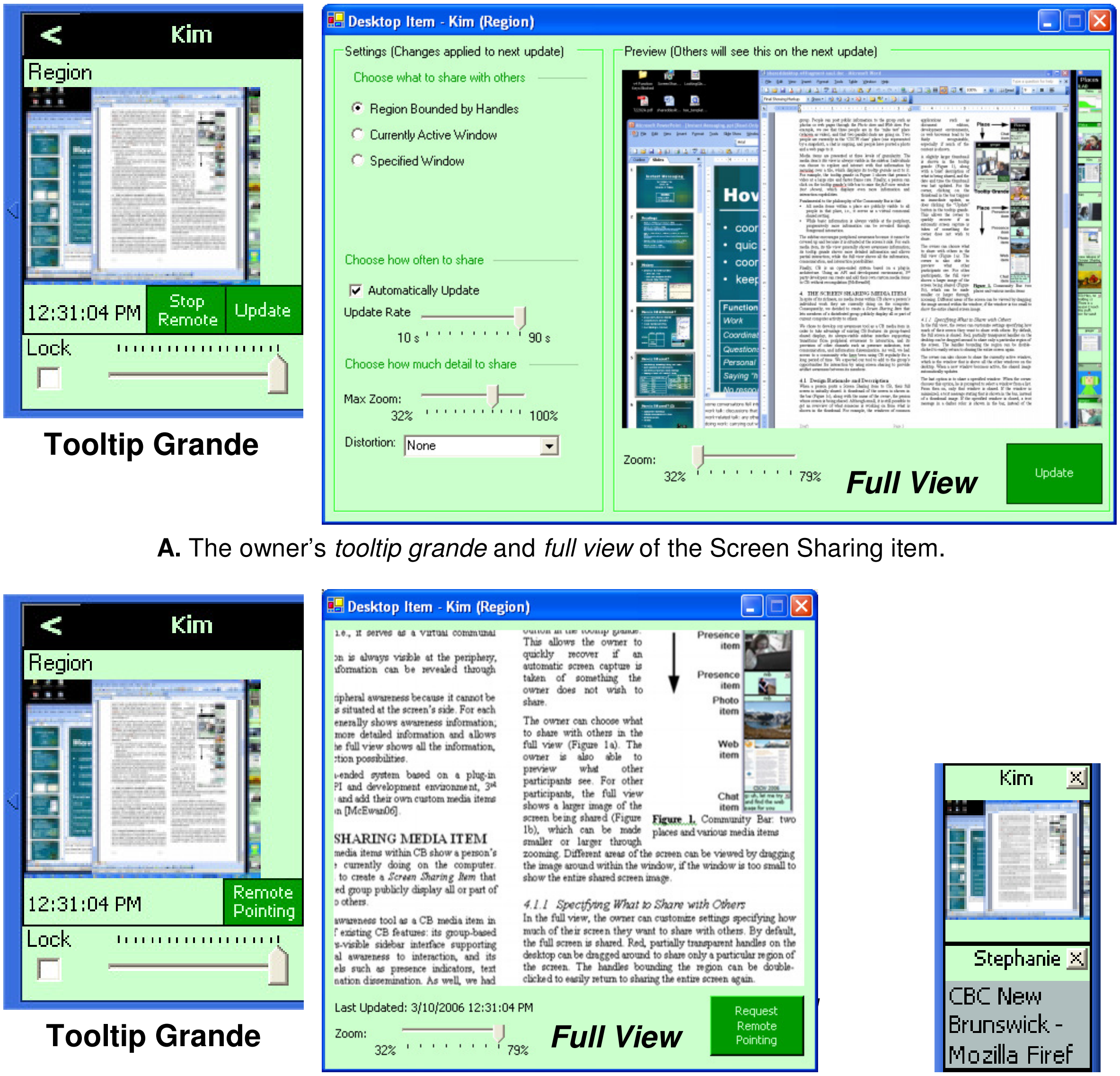
Figure 3.4 – Thumbnails of people’s screens.

working on the paper now, as well as the presentation (see my Shared Screen)” – this gives the group some context to help interpret the image.

By default, this thumbnail is updated once a minute. However, the owner (and only the owner) can trigger an immediate update by clicking the tile. For example, Kim may do this to rapidly replace a screen shot that she did not want others to see, or (more typically) to show others changed screen content in a timely way, e.g., as part of a discussion of the image that may be occurring in an ongoing text chat.

Although small, the thumbnail and its update frequency suffices to provide all others in that CB place with an overview of what Kim is working on. Typically, the visual characteristics of windows within the 70 x 60 pixel thumbnail are sufficiently recognizable (Kaasten et al., 2002) so that others can tell if the poster is editing a document, browsing the web, preparing a presentation, etc. While actual content is hard to distinguish, visual landmarks such as photos and text formatting are discernable. For example, the first five thumbnails in Figure 3.4 show people visiting a web page, editing a Word document with highlighting turned on, looking at email through the Outlook email reader and checking MSN Messenger’s buddy list, using the Picasso photo viewer, and working on a presentation. The last two thumbnails in the second row show particular regions of people’s screens: part of a code window in a programming environment, and part of a document.

Now reconsider the shared screen in the tile from Figure 3.1, shown again in Figure 3.5c. From the audience’s previous knowledge of Kim’s work habits and from the



Settings (Changes applied to next update)

Choose what to share with others

● Region Bounded by Handles
○ Currently Active Window
○ Specified Window

Choose how often to share

☑ Automatically Update

Update Rate 10 s – 90 s

Choose how much detail to share

Max Zoom: 32% – 100%

Distortion: None

Preview (Others will see this on the next update)

Zoom: 32% – 79%

i.e., it serves as a virtual communal

on is always visible at the periphery, iformation can be revealed through

ripheral awareness because it cannot be s situated at the screen's side. For each enerally shows awareness information; more detailed information and allows he full view shows all the information, ction possibilities.

-ended system based on a plug-in PI and development environment, 3rd and add their own custom media items n [McEwan06].

SHARING MEDIA ITEM

media items within CB show a person's currently doing on the computer. to create a *Screen Sharing Item* that ed group publicly display all or part of o others.

awareness tool as a CB media item in f existing CB features: its group-based s-visible sidebar interface supporting al awareness to interaction, and its els such as presence indicators, text nation dissemination. As well, we had

button in the tooltip grande. This allows the owner to quickly recover if an automatic screen capture is taken of something the owner does not wish to share.

The owner can choose what to share with others in the full view (Figure 1a). The owner is also able to preview what other participants see. For other participants, the full view shows a larger image of the screen being shared (Figure 1b), which can be made smaller or larger through zooming. Different areas of the screen can be viewed by dragging the image around within the window, if the window is too small to show the entire shared screen image.

Presence item
Presence item
Photo item
Web item
Chat item

Figure 1. Community Bar: two places and various media items

4.1.1 Specifying What to Share with Others

In the full view, the owner can customize settings specifying how much of their screen they want to share with others. By default, the full screen is shared. Red, partially transparent handles on the desktop can be dragged around to share only a particular region of the screen. The handles bounding the region can be double-clicked to easily return to sharing the entire screen again.

Last Updated: 3/10/2006 12:31:04 PM

Zoom: 32% – 79%

A. The owner's *tooltip grande* and *full view* of the Screen Sharing item.

B. The audience's *tooltip grande* and *full view* of the Screen Sharing item.

C. *Tile view*, all people

Figure 3.5 – Various views of the Screen Sharing item.

contents of her Chat item, they correctly guess that she is currently editing a document in Microsoft Word, where the Powerpoint presentation she mentions is partially visible in the background. The audience can also tell that this is a two-column document typical of most ACM papers, and that a figure is positioned at the top right of the page. If the viewer is a co-author of this paper, then that person could likely guess what page that is from their knowledge of the paper.

3.3.2 Tooltip Grande

The tooltip grande for both the owner and the audience shows a somewhat larger thumbnail (Figures 3.5a and 3.5b, left side). Akin to a glance, people in the CB place may raise this to help them further recognize certain features in the owner's screen. Above the thumbnail is a brief description of what is being shared, i.e., the full screen, a region of the screen, the active window, or a particular window (to be discussed shortly). Below the thumbnail is a timestamp indicating when the image was last updated. For example, the tooltip grande views of Kim's desktop in Figures 3.5a and 3.5b (left) show that she is sharing a region of the screen, and that it was last updated at 12:31 PM today.

The lock and slider that appear at the bottom of the tooltip grande are common to all CB media items, and are used by the viewer to adjust the size of the tile in the sidebar (McEwan and Greenberg, 2005). When the tile is resized to dimensions that are too small for the thumbnail to be recognizable, the thumbnail is replaced by a descriptive text label. To illustrate, the bottom tile in Figure 3.5c is a Screen Sharing item posted by Stephanie, where she is sharing a specific window ("CBC New Brunswick – Mozilla Firefox") rather than a screen region. Using its tooltip grande, the current viewer has shrunk his view of Stephanie's tile; only a text description is displayed that gives the name of the window being shared. Finally, the arrow at the top left of the tooltip grande is also common to all CB media items, and is used to invoke the full view, discussed shortly.

While the audience and owner view of the tooltip grande are visually similar, there are some differences. As before, the owner (and only the owner) can trigger an immediate update by clicking the thumbnail or by pressing the "Update" button visible at the tooltip grande's bottom right (Figure 3.5a, left). The audience view of the tooltip grande does not have this button, but the audience has a button in their view titled "Remote Pointing", while the owner has its corollary "Stop Remote". This will be explained later.

3.3.3 Full View

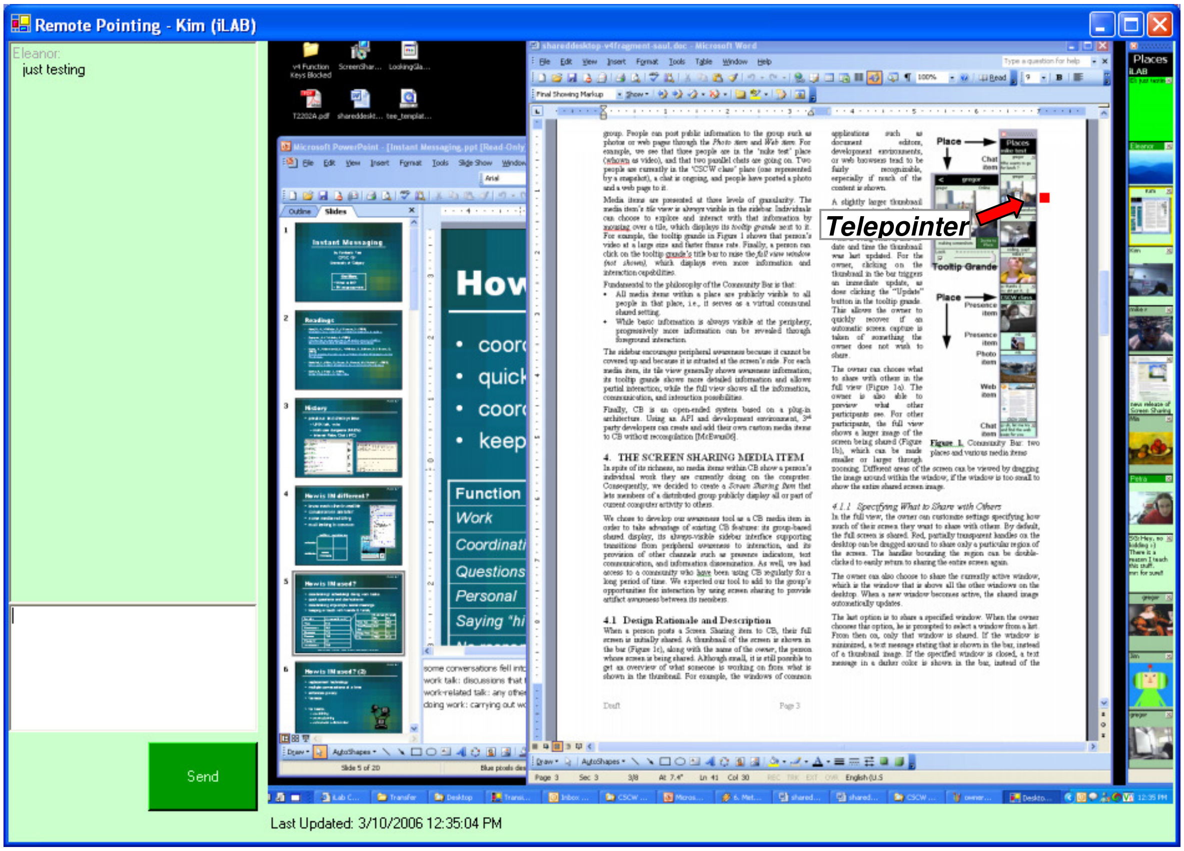
The full view as seen by both the audience and the owner gives a larger and much more detailed preview of the captured display (Figures 3.5a and 3.5b, right). As with other views, this preview is live: its contents are replaced as updates come in. As visible on the left side of the full view in Figure 3.5a, the owner has additional controls that will let him/her adjust and limit how the display is shared, thus providing some balance between awareness and privacy. These privacy controls are described in Section 3.4.

A zoom slider below the image lets the viewer zoom into the image as desired for greater detail. When the zoomed-in image does not fit within the window, the viewer can pan the image by directly selecting and dragging it with the mouse. As will shortly be discussed, for privacy reasons the permissible level of zoom depends on how the owner has configured sharing, i.e., zooming may be restricted to much less than true screen resolution. For example, in the full views in Figures 3.5a and 3.5b, it can be seen that Kim has set the maximum zoom level to 79%. The audience member is looking at her screen at this maximum zoom level (Figure 3.5b, full view), while Kim is looking at it at a 32% zoom so that the entire region fits the full view's window (Figure 3.5a, full view). It can also be seen that 79% zoom of the true screen resolution produces a fairly legible image (Figure 3.5b, full view): subsection titles of the paper are easily visible, and the paper text can be read with some effort.

3.3.4 Moving to Interaction

Any audience member can attempt to initiate a real-time remote pointing session. The owner sees a remote pointing request via a dialog box, and can approve or deny it. If the owner denies it, a short message is displayed to the audience member who requested remote pointing notifying them that permission was not granted.

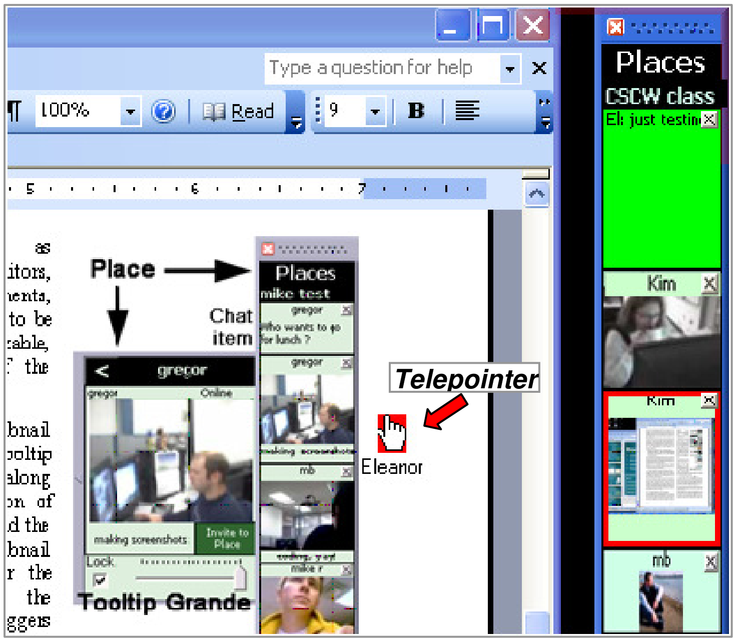
If the owner approves it, a remote pointing window (Figure 3.6a) appears on the screen of that audience member. This remote pointing window displays the shared screen image at the maximum allowable resolution. It also includes a full-sized chat box (Figure 3.6a, left) linked to a Chat item in the sidebar (Figure 3.6a and 3.6b, top right) so that the audience member can communicate with the owner in the same window as remote pointing, rather than having to switch between the remote pointing window and the CB sidebar or the Chat item's full view window.



Eleanor:
just testing

Last Updated: 3/10/2006 12:35:04 PM

A. Audience member view and telepointer control of another's desktop.



B. Partial screen snapshot of the owner's desktop, illustrating the telepointer. The red-outline in the Screen Sharing tile indicates that someone is looking at a full view of this screen.

Figure 3.6 – A remote pointing session.

The audience member can drag a small red telepointer around the shared screen image, visible at the top right of Figure 3.6a. A corresponding telepointer appears and moves around on the owner’s actual desktop in the corresponding location, as shown in the partial screenshot of the owner’s desktop in Figure 3.6b. Either the owner or the other participant can terminate the session at any time. Remote pointing is currently limited to two participants; if another person tries to request remote pointing while the owner is already involved in a remote pointing session, that person is notified of this and asked to try again later.

Although remote pointing is not as powerful as systems that let people take turns interacting with the application such as VNC (Richardson et al., 1998) or Timbuktu (WOS Data Systems, 1987), remote pointing suffices for most situations. As Whittaker suggests from his observations of casual interactions in offices, “Document use indicates a requirement for simple systems rather than full-blown shared editors. A system that allowed mutual viewing of documents, with the ability to point at and possibly make simple annotations, may be all that is required here” (Whittaker et al., 1994).

3.4 Privacy Controls

Privacy is, of course, a serious consideration in an always-on screen sharing system. For example, imagine a situation when Kim inadvertently displays a sensitive email message that others should not be seeing. The challenge is how people can balance the awareness information they want others to have of their work with their own privacy needs.

First and foremost, note that privacy is not just a technical issue (Boyle and Greenberg, 2005a). Rather, it is heavily dependant on the group culture and the actual practice of use that develops over time. As an always-on media space, Community Bar is designed for a community of intimate collaborators who have a real need and desire to stay connected. This is akin to a shared office of close-knit workers (or close friends, or family members) that are comfortable with seeing each other as they move around the shared space, as well as any information they are working on. Of course, this intended use could be abused by (say) an office manager that insists that all employees use the

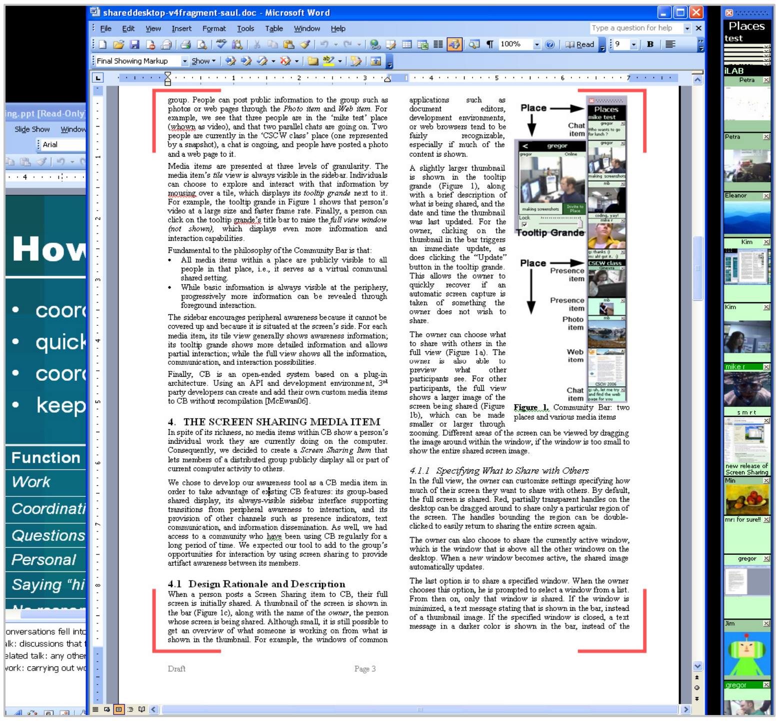
Screen Sharing media item so that their work can be monitored. However, even in the benign case, people may want some control over what others can see. Of course, the most restrictive control is to simply not show the item; this is the default, as the Screen Sharing media item only appears when created explicitly by the owner. Further controls and feedback offered by the Screen Sharing item are discussed in the following subsections.

3.4.1 Specifying What to Share with Others

Owners have full control over what to 'push' out as artifact awareness; the audience cannot 'pull' any extra information. Thus, the first level of privacy control is to let the owner specify how much of the display he or she wishes to share with others. The owner can choose what to share in the full view (Figure 3.5a, full view): a particular screen region selected by handles (which can include a small area up to and including the entire screen), the currently active window, or a particular user-specified window. The choice restricts what others can see to only those parts of the display the owner wishes to reveal.

By default, a screen region encompassing the full (primary) screen is shared. Anything displayed in this region is captured: partial and overlapping windows, background wallpaper, dialog boxes, etc. Semi-transparent red handles define the region being shared (Figure 3.7), and the owner can easily adjust the bounds of the region by dragging these handles around. For example, while Kim can share the entire display, she has repositioned the handles in Figure 3.7 to restrict sharing to the working area surrounding her text. She returns to sharing the entire (primary) screen by double-clicking a handle. For people with two displays, right-double-clicking a handle will share the full secondary screen.

The owner can also share individual windows. First, the owner can selectively share the currently active window. This is the window that has the input focus, and that appears atop all the other windows on the desktop. As the user switches to a different window (thus making that one the active window), the shared image automatically updates to replace the old window with this new one. Second, the owner can share a specified window from a list of all windows (Figure 3.8). When selected, only that



group. People can post public information to the group such as photos or web pages through the *Photo item* and *Web item*. For example, we see that three people are in the 'mike test' place (whown as video), and that two parallel chats are going on. Two people are currently in the 'CSCW class' place (one represented by a snapshot), a chat is ongoing, and people have posted a photo and a web page to it.

Media items are presented at three levels of granularity. The media item's *tile* view is always visible in the sidebar. Individuals can choose to explore and interact with that information by mousing over a tile, which displays its *tooltip grande* next to it. For example, the tooltip grande in Figure 1 shows that person's video at a large size and faster frame rate. Finally, a person can click on the tooltip grande's title bar to raise the *full view window (not shown)*, which displays even more information and interaction capabilities.

Fundamental to the philosophy of the Community Bar is that:

- All media items within a place are publicly visible to all people in that place, i.e., it serves as a virtual communal shared setting.
- While basic information is always visible at the periphery, progressively more information can be revealed through foreground interaction.

The sidebar encourages peripheral awareness because it cannot be covered up and because it is situated at the screen's side. For each media item, its tile view generally shows awareness information; its tooltip grande shows more detailed information and allows partial interaction; while the full view shows all the information, communication, and interaction possibilities.

Finally, CB is an open-ended system based on a plug-in architecture. Using an API and development environment, 3rd party developers can create and add their own custom media items to CB without recompilation [McEwan06].

4. THE SCREEN SHARING MEDIA ITEM

In spite of its richness, no media items within CB show a person's individual work they are currently doing on the computer. Consequently, we decided to create a *Screen Sharing Item* that lets members of a distributed group publicly display all or part of current computer activity to others.

We chose to develop our awareness tool as a CB media item in order to take advantage of existing CB features: its group-based shared display, its always-visible sidebar interface supporting transitions from peripheral awareness to interaction, and its provision of other channels such as presence indicators, text communication, and information dissemination. As well, we had access to a community who have been using CB regularly for a long period of time. We expected our tool to add to the group's opportunities for interaction by using screen sharing to provide artifact awareness between its members.

4.1 Design Rationale and Description

When a person posts a Screen Sharing item to CB, their full screen is initially shared. A thumbnail of the screen is shown in the bar (Figure 1c), along with the name of the *owner*, the person whose screen is being shared. Although small, it is still possible to get an overview of what someone is working on from what is shown in the thumbnail. For example, the windows of common applications such as document editors, development environments, or web browsers tend to be fairly recognizable, especially if much of the content is shown.

A slightly larger thumbnail is shown in the tooltip grande (Figure 1), along with a brief description of what is being shared, and the date and time the thumbnail was last updated. For the owner, clicking on the thumbnail in the bar triggers an immediate update, as does clicking the "Update" button in the tooltip grande. This allows the owner to quickly recover if an automatic screen capture is taken of something the owner does not wish to share.

The owner can choose what to share with others in the full view (Figure 1a). The owner is also able to preview what other participants see. For other participants, the full view shows a larger image of the screen being shared (Figure 1b), which can be made smaller or larger through zooming. Different areas of the screen can be viewed by dragging the image around within the window, if the window is too small to show the entire shared screen image.

Figure 1. Community Bar: two places and various media items

4.1.1 Specifying What to Share with Others

In the full view, the owner can customize settings specifying how much of their screen they want to share with others. By default, the full screen is shared. Red, partially transparent handles on the desktop can be dragged around to share only a particular region of the screen. The handles bounding the region can be double-clicked to easily return to sharing the entire screen again.

The owner can also choose to share the currently active window, which is the window that is above all the other windows on the desktop. When a new window becomes active, the shared image automatically updates.

The last option is to share a specified window. When the owner chooses this option, he is prompted to select a window from a list. From then on, only that window is shared. If the window is minimized, a text message stating that is shown in the bar, instead of a thumbnail image. If the specified window is closed, a text message in a darker color is shown in the bar, instead of the

Draft Page 3

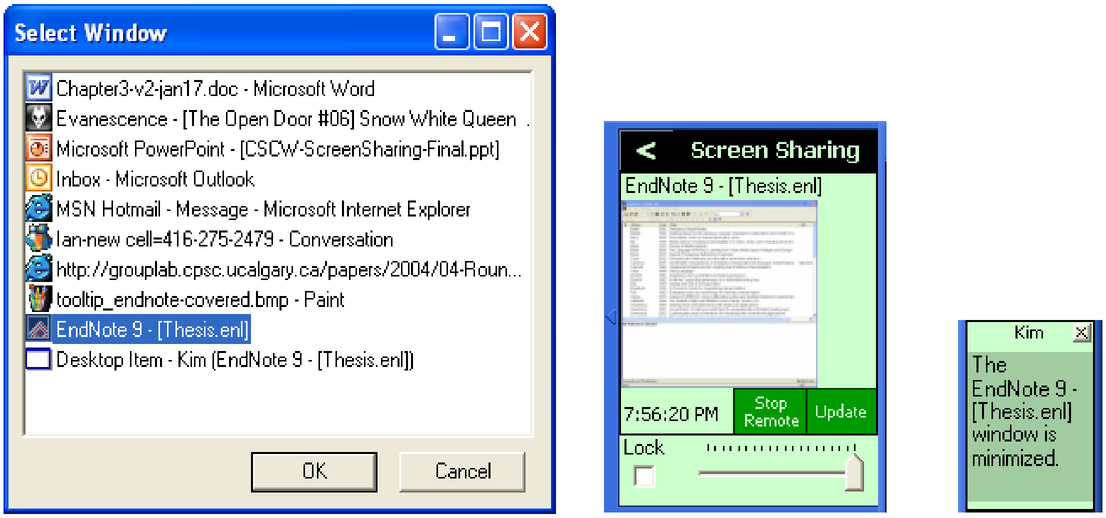
Figure 3.7 – The capture region of the screen is marked by adjustable semi-transparent red handles (enhanced here).

window is captured and shared (regardless of its position on the screen). If the owner minimizes or closes the window, an appropriate text message comprising the title of the window is shown instead of a thumbnail image (Figure 3.8c). When the owner resumes working in the window, the thumbnail is displayed again.

3.4.2 Specifying Update Frequency

The second level of privacy control lets the owner specify how often the display should be captured, and thus how often the audience gets this update. This control allows owners to reduce temporal fidelity.

The owner can also specify if updates are manual or automatic. If manual, the display is only updated when the owner clicks the thumbnail in the tile view, or the "Update" button in the tooltip grande and full view. If automatic, the owner can specify an update frequency interval between 10 and 90 seconds (although clicking the thumbnail/update button will take an immediate snapshot).


Select Window
Chapter3-v2-jan17.doc - Microsoft Word
Evanescence - [The Open Door #06] Snow White Queen .
Microsoft PowerPoint - [CSCW-ScreenSharing-Final.ppt]
Inbox - Microsoft Outlook
MSN Hotmail - Message - Microsoft Internet Explorer
Ian-new cell=416-275-2479 - Conversation
http://grouplab.cpsc.ucalgary.ca/papers/2004/04-Roun...
tooltip_endnote-covered.bmp - Paint
EndNote 9 - [Thesis.enl]
Desktop Item - Kim (EndNote 9 - [Thesis.enl])
OK
Cancel
Screen Sharing
EndNote 9 - [Thesis.enl]
7:56:20 PM
Stop Remote
Update
Lock
Kim

The EndNote 9 - [Thesis.enl] window is minimized.

A. Specifically sharing the *EndNote* window. **B.** Tooltip grande of the *EndNote* window. **C.** Tile view when the window is minimized.

Figure 3.8 – Sharing a specified window.

Unlike commercial screen sharing systems tailored for real-time interaction, this infrequent update should suffice for artifact awareness. These updates still inform others of basic activities while minimizing distraction and privacy concerns that might otherwise arise from real-time movement in the various views.

3.4.3 Specifying How Much Detail to Share

The third level of privacy control lets the owner manipulate the image fidelity that others can see. Techniques include adjusting the zoom level and distorting the image through image manipulation techniques. The less detail visible, the greater the perceived privacy.

An audience member is allowed to zoom into details in a shared screen image only up to a maximum zoom level set by the owner. Low zoom limits transform the image into a low resolution image. For example, if Kim set a low level zoom of around 33% and her captured region encompasses 1280 x 1024 pixels, the shared image is visually compressed to about 1/9 of the original area (~426 x 341). Alternatively, she can set an increasingly higher zoom limit, so that others can zoom in and view the shared image up to the original resolution. For example, Kim could set the level so that a viewer can read large-font section headings in a text document, but not the actual text contents in paragraphs.

Alternatively or in combination with zoom limits, the owner can mask and distort the image by selecting one of several image masking effects. Current options include image blurring, pixelization, and image randomization; others could be easily added. These distortion techniques offer people a high degree of control of image fidelity not only in the thumbnails, but in the larger zoomed-in full views as well.

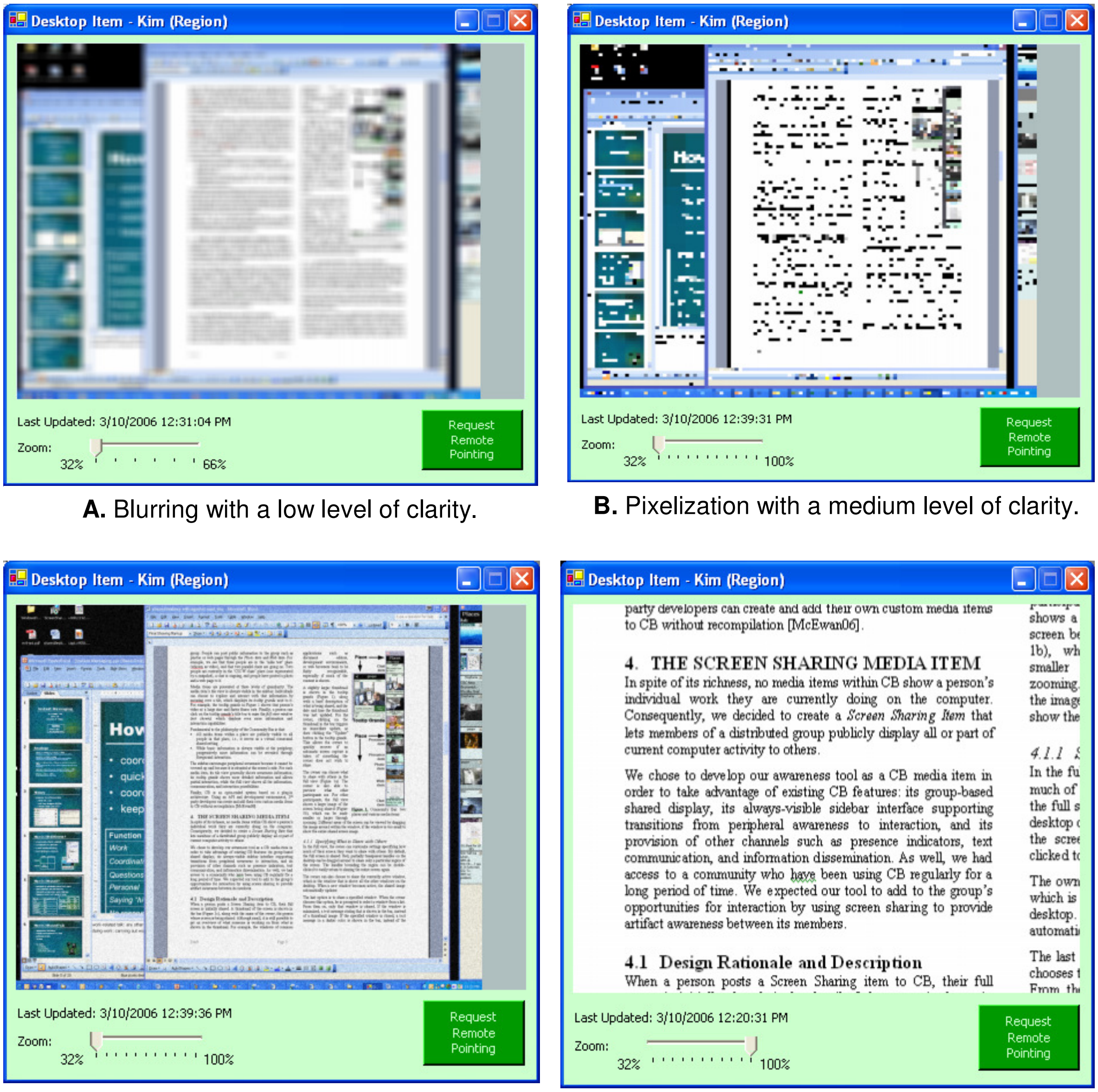
For example, Figure 3.9 illustrates what people would see when Kim uses the blur effect (3.9a), the pixelate effect (3.9b), or the randomization effect (3.9c) at 32% zoom. The first two let others roughly see what Kim is working on, while preserving her privacy because the image does not reveal legible detail. The last was not as useful as anticipated, and was later removed as an option from the Screen Sharing item. Figure 3.9d shows that people would see Kim's screen at the original resolution when using 100% zoom with no distortion; what Kim sees, other people can see too.

3.4.4 Feedback of Image Capture

The fourth level of privacy control is for the system to provide sufficient feedback to the owner about what others can see. First, the owner can always see exactly what the audience can see, because the Screen Sharing item is visible on both owners and audience members' sidebars. Similarly, if the owner raises the tooltip grande or the full view, they see the same image as the audience member.

Second, several mechanisms warn the owner just before an auto-update happens. Five seconds prior to an auto-update, the owner's tile in the bar is outlined in yellow. If sharing a region, the red handles that bound the region turn yellow as well. Colours revert back to normal after the update is completed. This feedback aims to be a reasonable compromise that reminds the owner of what is going on without being overly distracting.

Third, when an audience member opens the full view belonging to the owner, the Screen Sharing item in the owner's bar is outlined in red (Figure 3.6b). This outline remains until the full view is closed. However, no identifying information is supplied as to who is looking at the full view; it acts only as an indication that at least one person has the full view open.



Desktop Item - Kim (Region)

Last Updated: 3/10/2006 12:31:04 PM

Zoom: 32% 66%

A. Blurring with a low level of clarity.

Desktop Item - Kim (Region)

Last Updated: 3/10/2006 12:39:31 PM

Zoom: 32% 100%

B. Pixelization with a medium level of clarity.

Desktop Item - Kim (Region)

Last Updated: 3/10/2006 12:39:36 PM

Zoom: 32% 100%

C. Randomization with a low level of clarity.

Desktop Item - Kim (Region)

party developers can create and add their own custom media items to CB without recompilation [McEwan06].

4. THE SCREEN SHARING MEDIA ITEM

In spite of its richness, no media items within CB show a person's individual work they are currently doing on the computer. Consequently, we decided to create a *Screen Sharing Item* that lets members of a distributed group publicly display all or part of current computer activity to others.

We chose to develop our awareness tool as a CB media item in order to take advantage of existing CB features: its group-based shared display, its always-visible sidebar interface supporting transitions from peripheral awareness to interaction, and its provision of other channels such as presence indicators, text communication, and information dissemination. As well, we had access to a community who have been using CB regularly for a long period of time. We expected our tool to add to the group's opportunities for interaction by using screen sharing to provide artifact awareness between its members.

4.1 Design Rationale and Description

When a person posts a Screen Sharing item to CB, their full

shows a
screen be
1b), wh
smaller
zooming.
the image
show the

4.1.1

In the fu
much of
the full s
desktop
the scree
clicked to

The own
which is
desktop.
automati

The last
chooses t
From th

Last Updated: 3/10/2006 12:20:31 PM

Zoom: 32% 100%

D. Full zoom with no distortion.

Figure 3.9 – Various masking effects and zoom levels applied to the shared screen image.

3.4.5 Communal Feedback

The fifth level of privacy control is social, as defined by the CB group. Screen Sharing items are visible only to the other people in the CB place. Because all people logged on in a CB place are visible, the owner of an item knows who can see. As well, because people in a CB place are part of a social group, one can reasonably expect – security violations aside – that only socially appropriate people can see it. Finally, because these people are expected to be colleagues, the viewers themselves can use the other facilities in the Community Bar to warn the owner about inappropriate things that are being shared.

3.5 Summary

In this chapter, I have presented a prototype of an artifact awareness tool that uses screen sharing to provide information about other people's artifacts and individual work. With the Screen Sharing item, people see others' screens in miniature at the edge of their display, can selectively raise a larger view of that screen to get more detail, and can engage in remote pointing if desired. People can balance awareness with privacy, by using the privacy-protection strategies built into the system: restricting what parts of the screen others can see, specifying update frequency, hiding image detail, and getting feedback of when screenshots are taken.

The Screen Sharing item was built within the context of the Community Bar, an augmented media space, and it exploits Community Bar's existing features for supporting informal awareness and casual interaction. The Screen Sharing item is different from other Community Bar media items in that it was designed specifically to support awareness of people's artifacts and individual work. Also, once posted, the Screen Sharing item shares the activities that people are currently engaged in on their computers without them having to explicitly post each artifact that they are working with and would like to share. By using the Screen Sharing item, people can get a better sense of what others are working on and whether they are available for conversation or collaboration. Of course, the Screen Sharing item in Community Bar is only one example of how artifact awareness could be provided to a distributed group through screen sharing; other alternative interface designs are briefly discussed in Section 7.3.3.

In the following chapter, I describe the architecture and implementation details behind the Screen Sharing item. In Chapter 5, I discuss the initial reactions and experiences that people had when using the Screen Sharing item. In Chapter 6, I present a theoretical analysis of the Screen Sharing item's effect on privacy.

Chapter 4. Implementation Details

In the preceding chapter, I introduced the Screen Sharing item artifact awareness tool and showed how it fit within the context of Community Bar. In this chapter, I describe its implementation details and discuss the rationale behind its technical design. I start by first giving an overview of Community Bar's architecture and extensible design, which allow third-party developers to add their own media items to Community Bar as plug-ins. I then show how the Screen Sharing item was implemented as an external plug-in to fit within the Community Bar. Finally, I report on results from performance tests that suggest that the Screen Sharing item suffices for its expected use and can lead to a satisfactory end-user experience, and I also briefly discuss the areas revealed by the performance tests in which the Screen Sharing item's performance could be improved.

4.1 Extending Community Bar

Section 3.2 previously introduced Community Bar (CB) as a media space that can be augmented with extra information to support informal awareness and casual interaction in distributed groups. Individual members can broadcast and receive information about themselves and others, and latecomers can join and synchronize their information with the rest of the group's. However, CB is also a toolkit that lets third-party developers add their own media items to CB; this is what I did with the Screen Sharing item artifact awareness tool. In this section, I give a brief overview of how CB and its extensible architecture work.

4.1.1 Community Bar's Architecture

CB follows a client/server architecture and uses a distributed model-view-controller (dMVC) pattern combined with a notification engine (McEwan et al., 2006), as illustrated

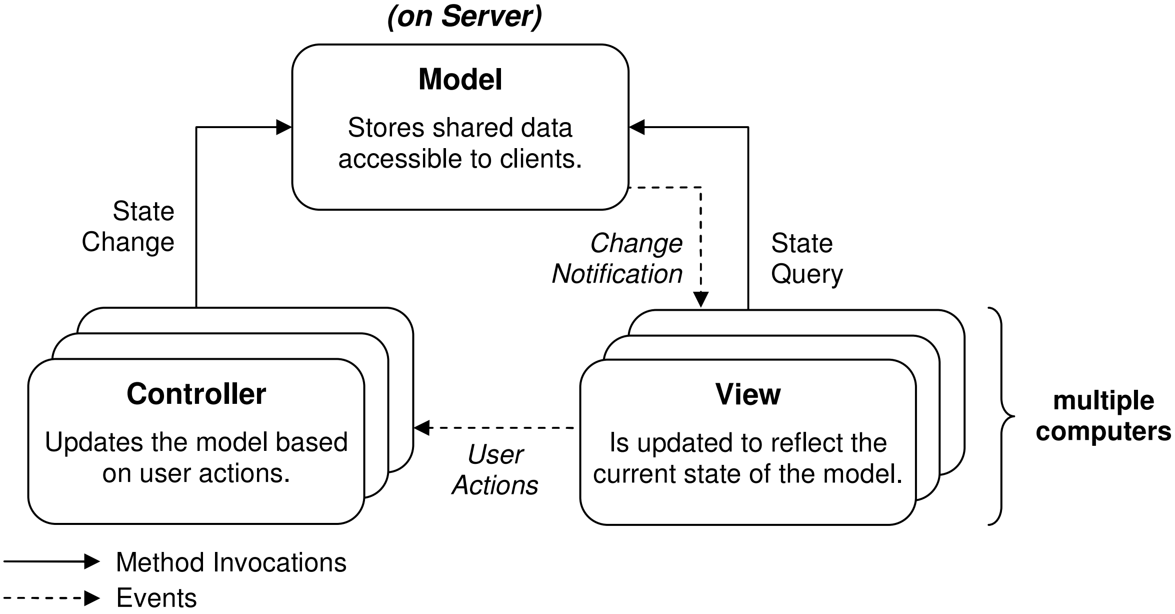
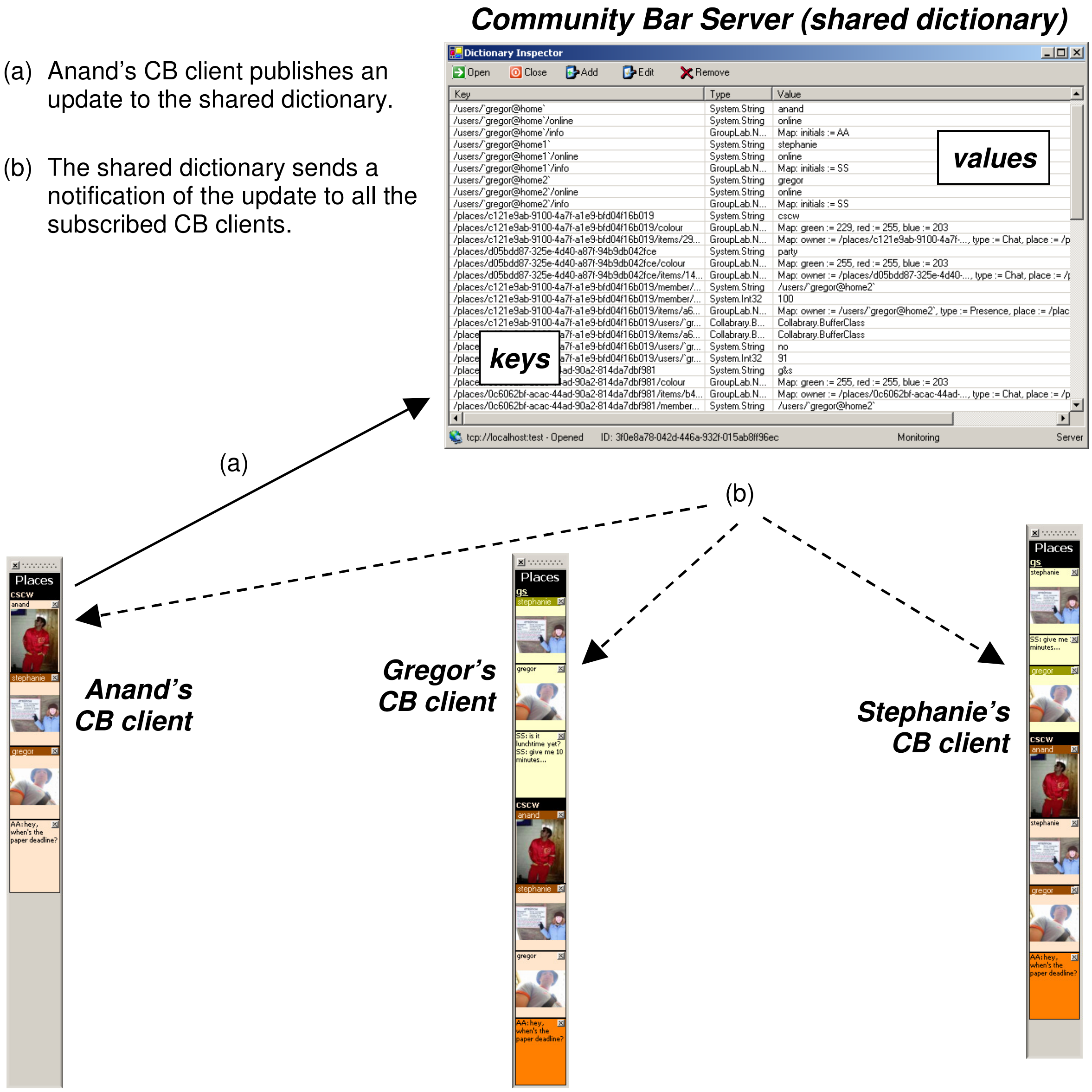

(on Server)
Model
Stores shared data accessible to clients.
State Change
Change Notification
State Query
Controller
Updates the model based on user actions.
User Actions
View
Is updated to reflect the current state of the model.
multiple computers
Method Invocations
Events

Figure 4.1 – Distributed model-view-controller pattern.

in Figure 4.1. Basically, a central server hosts a persistent data store, the *model* (Figure 4.1, top), which CB clients can access to get shared information. This model is usually updated in response to user actions, by one or more *controllers* within the various CB clients (Figure 4.1, bottom left). When changes have been made to the model, the server generates *notifications* to tell the CB clients that their information is stale. The CB clients then update their *views* to reflect the current state of the model (Figure 4.1, bottom right).

To implement dMVC, CB uses GroupLab.Networking (Boyle, 2005; Boyle and Greenberg, 2005b), a toolkit for prototyping distributed multimedia groupware. GroupLab.Networking provides a *shared dictionary* server (Figure 4.2, top) that stores data in a hierarchical structure in which string *keys* are mapped to *values* of any type. This shared dictionary server stores the model for CB and its media items. Since the model is stored in a centralized repository, latecomers who connect to CB after a session has started can easily update their view to the shared state. Shared state is also synchronized between all the CB clients.

Figure 4.2 (modified from McEwan, 2006), illustrates how CB makes use of the shared dictionary to broadcast data. Controllers in the CB clients *publish* an update to the model by adding, deleting, or modifying keys and their values in the shared dictionary



(a) Anand's CB client publishes an update to the shared dictionary.

(b) The shared dictionary sends a notification of the update to all the subscribed CB clients.

Key	Type	Value
/users/`gregor@home`	System.String	anand
/users/`gregor@home`/online	System.String	online
/users/`gregor@home`/info	GroupLab.N...	Map: initials := AA
/users/`gregor@home1`	System.String	stephanie
/users/`gregor@home1`/online	System.String	online
/users/`gregor@home1`/info	GroupLab.N...	Map: initials := SS
/users/`gregor@home2`	System.String	gregor
/users/`gregor@home2`/online	System.String	online
/users/`gregor@home2`/info	GroupLab.N...	Map: initials := SS
/places/c121e9ab-9100-4a7f-a1e9-bfd04f16b019	System.String	cscw
/places/c121e9ab-9100-4a7f-a1e9-bfd04f16b019/colour	GroupLab.N...	Map: green := 229, red := 255, blue := 203
/places/c121e9ab-9100-4a7f-a1e9-bfd04f16b019/items/29...	GroupLab.N...	Map: owner := /places/c121e9ab-9100-4a7f-..., type := Chat, place := /p
/places/d05bdd87-325e-4d40-a87f-94b9db042fce	System.String	party
/places/d05bdd87-325e-4d40-a87f-94b9db042fce/colour	GroupLab.N...	Map: green := 255, red := 255, blue := 203
/places/d05bdd87-325e-4d40-a87f-94b9db042fce/items/14...	GroupLab.N...	Map: owner := /places/d05bdd87-325e-4d40-..., type := Chat, place := /p
/places/c121e9ab-9100-4a7f-a1e9-bfd04f16b019/member/...	System.String	/users/`gregor@home2`
/places/c121e9ab-9100-4a7f-a1e9-bfd04f16b019/member/...	System.Int32	100
/places/c121e9ab-9100-4a7f-a1e9-bfd04f16b019/items/a6...	GroupLab.N...	Map: owner := /users/`gregor@home2`, type := Presence, place := /plac
/places/c121e9ab-9100-4a7f-a1e9-bfd04f16b019/users/`gr...	Collabrary.B...	Collabrary.BufferClass
/place a7f-a1e9-bfd04f16b019/items/a6...	Collabrary.B...	Collabrary.BufferClass
/place a7f-a1e9-bfd04f16b019/users/`gr...	System.String	no
/place a7f-a1e9-bfd04f16b019/users/`gr...	System.Int32	91
/place ad-90a2-814da7dbf981	System.String	g&s
/place ad-90a2-814da7dbf981/colour	GroupLab.N...	Map: green := 255, red := 255, blue := 203
/places/0c6062bf-acac-44ad-90a2-814da7dbf981/items/b4...	GroupLab.N...	Map: owner := /places/0c6062bf-acac-44ad-..., type := Chat, place := /p
/places/0c6062bf-acac-44ad-90a2-814da7dbf981/member...	System.String	/users/`gregor@home2`

tcp://localhost:test - Opened   ID: 3f0e8a78-042d-446a-932f-015ab8ff96ec   Monitoring   Server

Figure 4.2 – A Community Server with three Community Bar clients connected (modified from McEwan, 2006). The server view shows the listing of the key-value pairs in the shared dictionary.

(Figure 4.2a) in response to user actions and/or other forms of input, e.g., new frames arriving from a web camera. The shared dictionary accepts *subscriptions* from clients who would like to be notified when a particular value in the shared dictionary changes, so that whenever changes are made, notifications are only sent to the clients that are interested in them (Figure 4.2b), i.e., for the purpose of updating their views.

4.1.2 Adding Custom Media Items

To add custom media items to CB, developers implement a C# code interface provided by CB in order to produce a Dynamic Link Library (DLL), the plug-in, which CB loads at runtime. Through this interface, developers have access to the shared dictionary and can add, remove, or modify data in it as well as subscribe to its data. The interface is used to specify how the item should be started, stopped, and displayed for its different views, i.e., its tile, tooltip grande, and separate views as illustrated in Figure 3.5.

Because of the interface, the plug-in architecture, and the dMVC for distributed data management, developers do not need access to CB's source code in order to create their own media items. Instead, they only need to specify how their items' views will receive and display information and how controllers will publish information in response to user actions and other forms of input. In the next section, I describe how the Screen Sharing item was implemented as a custom CB media item.

4.2 The Screen Sharing Item

As mentioned in the previous chapter, the Screen Sharing item was developed as a custom CB media item in order to leverage CB's existing features for supporting group awareness and communication. Consequently, the Screen Sharing item plugs into CB's client/server architecture and follows CB's dMVC data distribution model. As a result, it also uses a centralized approach to implement image-based screen sharing. The Screen Sharing item was written in C# using Visual Studio 2003 and runs (only) on Windows machines that have CB installed. More details on how the Screen Sharing item was implemented are given in the following sections.

4.2.1 Screen and Window Sharing Details

As mentioned in Section 2.3.3, the Screen Sharing item follows a centralized, image-based approach to screen sharing. There were several good reasons for using this approach:

- the awareness tool needed to be able to provide different-sized views of each screen or window being shared so an image-based approach was particularly appropriate since images are easily cropped and resized as needed;
- an image-based approach to sharing screens was reasonably easy and efficient to implement, partly because of the centralized way CB and its media items work;
- while the method used for sending shared screen image updates was fairly crude, e.g., see Section 2.3.3 for better methods, it was also 'good enough' in terms of performance (discussed in Section 4.3).

Basically, an owner's instance of the Screen Sharing item acts as a manager in a screen sharing system (see Section 2.3.3 for more information on managers) while also acting as a controller in CB's dMVC data distribution model (see Section 4.1.1 for details on CB and dMVC).

To share an owner's screen with other audience members, the Screen Sharing item first captures the region of the screen to share (Figure 4.3, Step 1). Next, it applies any image distortions the owner uses (Figure 4.3, Step 2) and optimizes the shared screen image for transmission across the network (Figure 4.3, Step 3). It then sends the shared screen image to the shared dictionary server (Figure 4.3, Step 4), which notifies the audience members' Screen Sharing items of the change (Figure 4.3, Step 5). Each step in this process is described in more detail below.

Step 1: Capturing the screen region to share

By default, the Screen Sharing item shares the owner's full screen. The owner can share a smaller region of the screen by moving handles around; the Screen Sharing item only captures the area of the screen that is bounded by the handles, which are actually transparent windows always on top of other windows.

Alternatively, the owner can choose to share only the currently active window, i.e., the window that is currently on top of all other windows. The Screen Sharing item keeps track of which window is on top by monitoring mouse clicks across different applications using the WindowsHooker library [grouplab.cpsc.ucalgary.ca/cookbook]. When a mouse

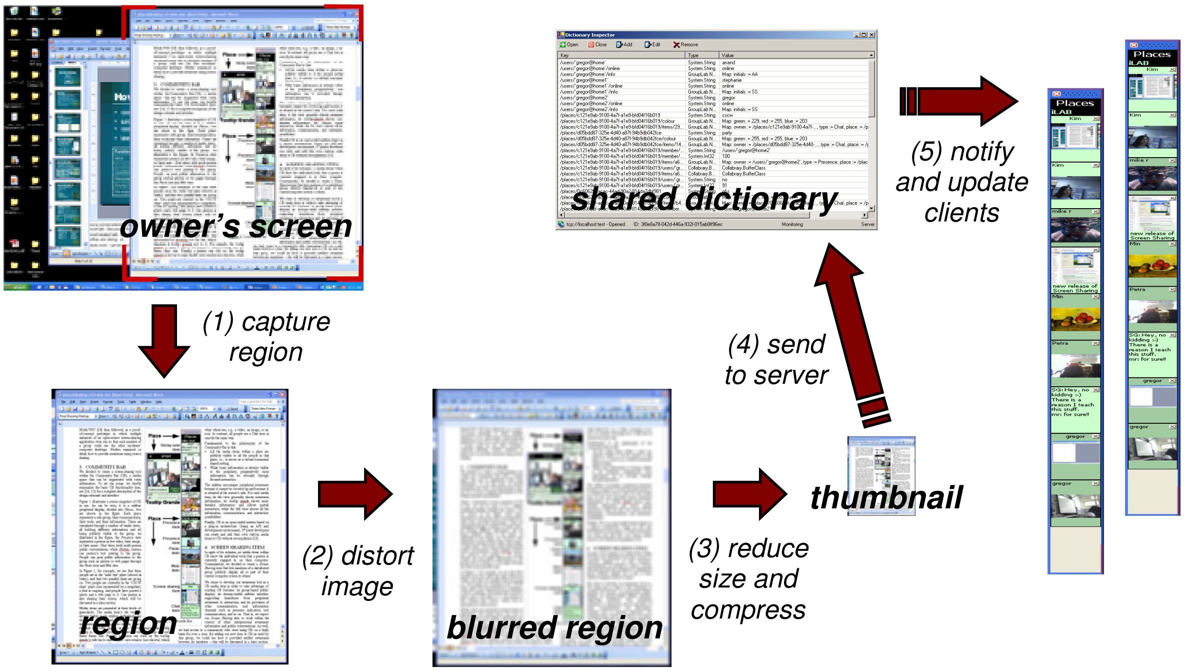
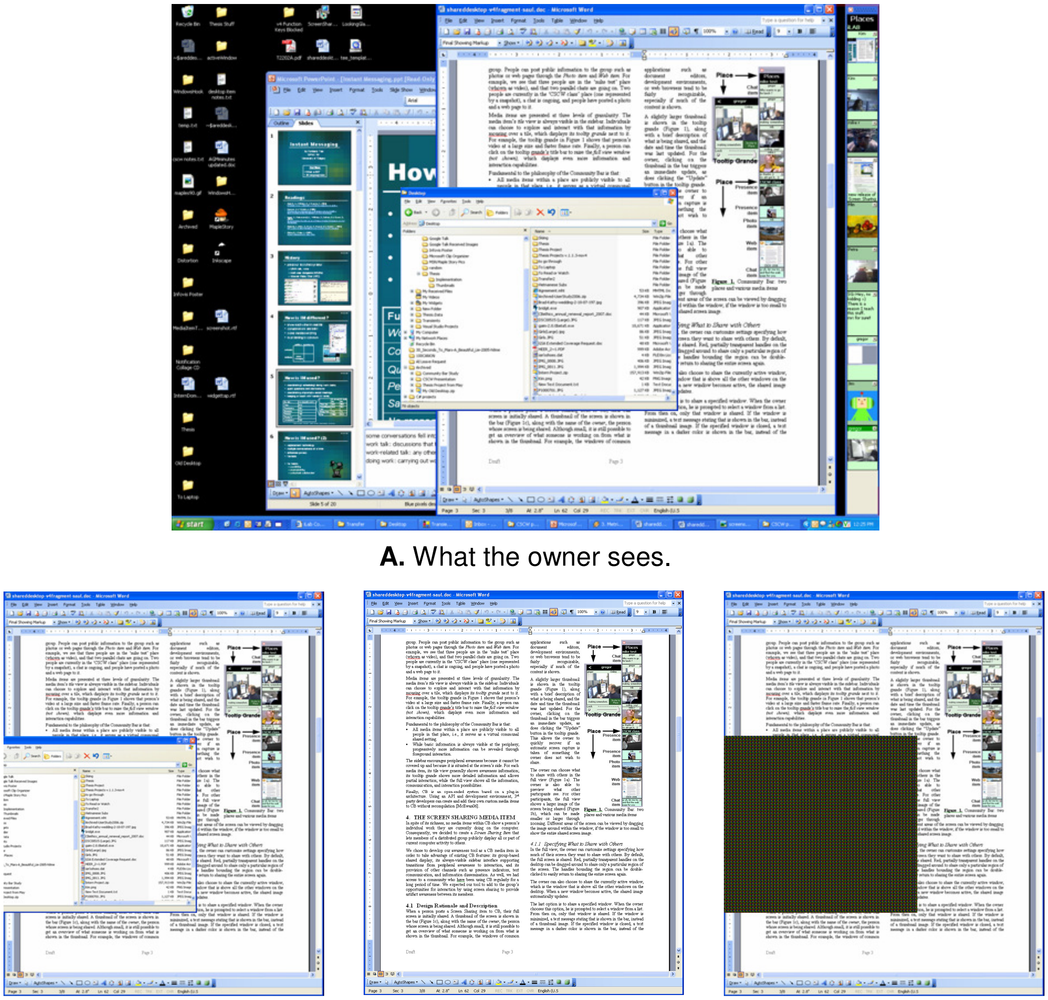


Figure 4.3 – The process of sharing a screen in CB using the Screen Sharing item.

click is detected, the Screen Sharing item determines which application is active by retrieving the main window handle of the child window or control that was clicked on. This application window is then captured and shared by the Screen Sharing item until a different window is activated. While there is a small overhead each time a mouse click happens (since a check must be made for whether the currently active window has changed), the effect on system performance seems to be negligible. Perhaps a slightly more efficient way to determine the currently active window would have been to hook directly into window events, but neither WindowsHooker nor the .NET Framework supported this (and it would have been onerous to implement from scratch).

Another option for the owner is to share only a particular window. The Screen Sharing item detects the windows currently open and allows the owner to select one to share from a list (Figure 3.8). This window is then captured and shared until a different option is chosen. If the specified window is minimized or closed, the Screen Sharing item displays an appropriate message in the sidebar in place of the thumbnail image. If the specified window is overlapped by other windows, the Screen Sharing item captures and shares the regions of those windows on top of the specified window (Figure 4.4b), since



A. What the owner sees.

B. What the audience currently sees.

C. Ideally, what the audience would see.

D. Alternative option for what the audience could see.

Figure 4.4 – Sharing a specified window overlapped by another window.

the Screen Sharing item simply captures the region of the screen bounded by the specified window's bounds. A previous approach to capturing only the specified window involved temporarily bringing it in front of the other windows, taking a snapshot of it, and then putting it back (Figure 4.4c). However, this approach caused an irritating 'flicker' each time it was done, so was not used in later versions of the system.

Alternatively, regions of the window where overlap occurs could be blocked out (Figure 4.4d). This approach is taken by some application sharing programs such as NetMeeting in order to prevent people from accidentally sharing sensitive information, and could be easily implemented in the Screen Sharing item. In any case, the Screen

Sharing item periodically (or on-demand, if being updated manually) captures a snapshot of the screen region or window that the owner is sharing (Figure 4.3, left).

Step 2: Distorting the shared screen image

Next, any distortions such as blurring or pixelization (Figure 3.9) are applied to the image (Figure 4.3, middle left). Distorting the shared screen image before it is sent to the shared dictionary ensures that audience members have no way to access a full-fidelity version of the shared screen image if the owner does not wish to share it. The Screen Sharing item uses distortion filters from the Collabrary library [grouplab.cpsc.ucalgary.ca/cookbook].

Step 3: Optimizing the shared screen image for network transmission

Finally, before being sent to the shared dictionary on the server, a couple of basic optimizations are done to the shared screen image to minimize the amount of data sent across the network (Figure 4.3, middle right).

First, the shared screen image is scaled down to a thumbnail of 150 x 150 pixels. Most of the time, people only look at shared screens in the tile or tooltip grande view where only a thumbnail of the shared screen is needed. If an audience member wants to see a larger view of the shared screen, e.g., in the separate window, a message is sent to the owner's Screen Sharing item to send a larger version of the shared screen image to the shared dictionary so that the CB client can download it from there. While this adds a perceptible but small delay to the time it takes to display the larger view the first time that it is needed (if no other audience member is also looking at the larger view), subsequent updates of the larger view are made along with the thumbnail updates without this delay.

Second, the thumbnail of the shared screen image is compressed in PNG (Portable Network Graphics) format. Although PNG files tend to be larger than JPEG files, PNG files preserve image quality, which was important for the Screen Sharing item since shared screens and windows tend to contain a lot of text.

While these optimizations were fairly basic, they generally suffice for expected use (Section 4.3). See Section 2.3.3 and Section 4.3.4 for additional methods that could be incorporated into the Screen Sharing item to improve performance even more.

Step 4: Sending the shared screen image to the server

The shared screen image and information related to it are the primary pieces of data stored on the server by the Screen Sharing item, although other information required for remote pointing and privacy controls is also stored there. Section 4.2.3 will describe the more important keys and values that the Screen Sharing item stores in the shared dictionary.

Step 5: Notifying and updating CB clients

Whenever a shared screen image has been updated in the shared dictionary on the server, notifications are sent out to the subscribed CB clients informing them of the change. The CB clients then retrieve the new shared screen image from the shared dictionary and update their views to the audience (Figure 4.3, top right). As the audience interacts with the shared screen in the different views, i.e., the tile, tooltip grande, and separate views, the information is presented differently, yet the source data from the model does not change (except for the full size vs. thumbnail images).

4.2.2 Remote Pointing Details

If the owner grants permission for remote pointing, a remote pointing window (Figure 3.6a) appears on the screen of the audience member who requested remote pointing. This window contains an area where the shared screen image is displayed at the maximum allowable resolution, i.e., the size that the owner has set as the maximum percentage of the original screen image that others can zoom to. This window also contains a chat box that is linked to a Chat item in the sidebar. The first time a remote pointing window is opened, the Screen Sharing item creates a new Chat item. The Screen Sharing item remembers this Chat item's shared dictionary key (Table 4.1, line 11) so that if the same audience member requests remote pointing again, the Chat item can be reused.

Whenever the audience member moves the telepointer, the Screen Sharing item calculates the telepointer's coordinates as a percentage of the displayed area. These coordinates are then stored in the shared dictionary (Table 4.1, line 13) so that the owner's Screen Sharing item can retrieve them, transform them into screen coordinates, and move the audience member's pointer on the display accordingly. The next section contains more details about exactly what is stored in the shared dictionary.

4.2.3 Data Stored in the Shared Dictionary

Table 4.1 gives some examples of the specific keys and values stored in the shared dictionary by the Screen Sharing item. Besides the shared screen image (Table 4.1, lines 1-2), some descriptive information is also stored such as the date/time the image was captured, a few keywords of what is being shared, any messages such as whether a shared window has been closed, an indicator for whether a person has been idle for five minutes or more, and a count for the number of audience members in the group that have requested a large version of the shared screen image (Table 4.1, lines 3-7). For a remote pointing session, the shared dictionary is used to transfer information such as the name of the person requesting remote pointing and whether or not the owner granted permission for the session to start (Table 4.1, lines 9-10). The shared dictionary also transfers information about what is being pointed at and who is doing the pointing (Table 4.1, lines 12-13).

Most of the data stored in the shared dictionary is not persistent; the basic information related to a screen sharing session (Table 4.1, lines 1-7) or a remote pointing session (Table 4.1, lines 8-14) is removed when the session is completed or the owner logs off of CB. However, owners' privacy settings (Table 4.1, lines 15-19) are kept in the shared dictionary from one session to the next, so that people do not need to reconfigure their settings each time they share their screen. The shared dictionary stores settings such as whether the shared screen image should be automatically updated, and if so, how often it should be updated (Table 4.1, lines 15-16). It also stores what the maximum image size can be for audience members, whether any distortion is involved, and if so, how much distortion to apply (Table 4.1, lines 17-19).

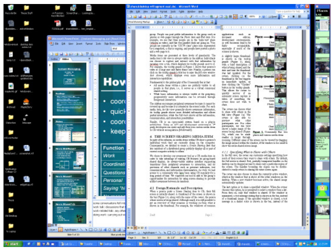
#	Shared Dictionary Key	Stored Data
Screen Sharing		
1	/items/<item_GUID>/desktopsmall	Thumbnail = (Collabrary.BufferClass) 
2	/items/<item_GUID>/desktopbig	Large Shared Screen Image = (Image)
3	/items/<item_GUID>/lastupdated	DateTime = “7/1/2007 5:16:18 PM”
4	/items/<item_GUID>/desc	Description = “Primary Screen”
5	/items/<item_GUID>/imageerrmsg	Error Msg = <null>
6	/items/<item_GUID>/idlestatus	Is Idle? = False
7	/items/<item_GUID>/desktoprequestbig	No. of Requests for Large Image = 1
Remote Pointing		
8	/items/<item_GUID>/permission/nostore	Attribute = True
9	/items/<item_GUID>/permission	Person Requesting = “Saul”
10	/items/<item_GUID>/granted	Permission Granted? = True
11	/items/<item_GUID>/chatitem	Key of Chat Item = “/items/<chat_GUID>”
12	/items/<item_GUID>/desktoppointer	Person Pointing = “Saul”
13	/items/<item_GUID>/desktoppointing	Point = (X=38,Y=24)
14	/items/<item_GUID>/stopremote	Person Stopping = <null>
Persistent User Settings		
15	users/<user_ID>/desktop_autoupdate	Automatically Update? = True
16	users/<user_ID>/desktop_updaterate	Update Rate = 6
17	users/<user_ID>/desktop_maxpercent	Maximum Percent = 75
18	users/<user_ID>/desktop_distortion	Distortion = Blur
19	users/<user_ID>/desktop_clarity	Clarity Level = -1

Table 4.1 – Sample keys and values stored in the shared dictionary by the Screen Sharing item.

4.3 Performance Tests for the Screen Sharing Item

As a research prototype, the Screen Sharing item was implemented to be robust enough to deploy for use so that the effects of screen sharing for artifact awareness could be studied. The Screen Sharing item does not implement advanced performance optimizations such as those described in Section 2.3.3. However, I show in this section how the performance of the Screen Sharing item is typically ‘good enough’ for its expected use.

4.3.1 Method

For the performance tests, ‘expected use’ of the Screen Sharing item was broken down into two main components: peripheral awareness (updates when the shared screen image does not have the user’s full attention) and focused interaction (real-time discussion or collaboration over artifacts seen in the shared screen). These components were further broken down into the five steps taken to share an owner’s screen with other audience members (discussed in Section 4.2.1): capturing the screen, distorting the screen image, resizing the screen image to be sent to the shared dictionary, sending the screen image to the shared dictionary, and retrieving the screen image from the shared dictionary.

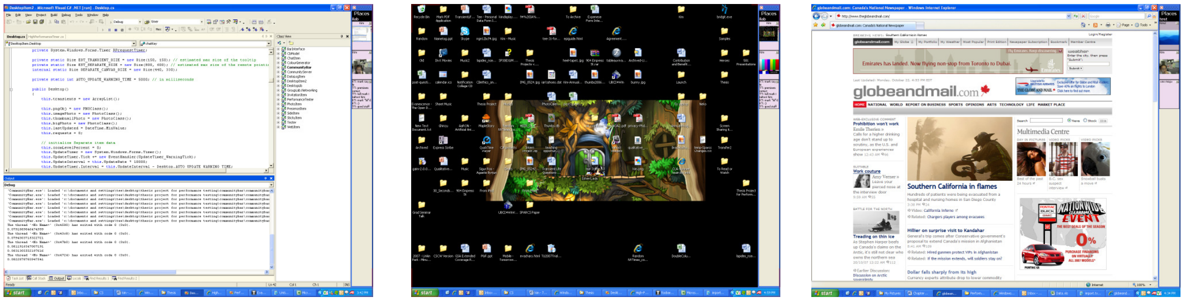
Code for measuring the speed of the first three steps was inserted into the code for the Screen Sharing item. Code for measuring the size of the data being sent to the shared dictionary was also inserted into the code for the Screen Sharing item. The Screen Sharing item was set up to use an auto-update interval of 10 seconds with blur at half clarity. For each condition (see the following section for factors), fifteen consecutive Screen Sharing item updates were measured. This process was repeated twice at different times of the day so that varying numbers of applications were open on the machines being tested.

The results from these measurements, together with results from network speed tests run (detailed in Section A.3), were then used to calculate the speed of the last two steps. Finally, the individual times from each of the steps were added together to give an approximation of the average total time taken to update a shared screen image. Summary statistics are reported in this chapter and additional detail can be found in Appendix A.

4.3.2 Factors

In the performance tests, the following three factors were used:

- computer (*Computer A*, *Computer B*): ‘Computer A’ is a Pentium 4, 3.4 GHz computer with 2GB of RAM; ‘Computer B’ is a Xeon, 1.4 GHz computer with 512 MB of RAM.



A. Code **B.** Desktop **C.** Web

Figure 4.5 – Example screens captured in the performance tests.

- capture area (*1024 x 768*, *1280 x 1024*, *1200 x 1600, dual monitor – 1280 x 1024 plus 1200 x 1600*): Full screens were captured for the performance tests since the Screen Sharing item shares the owner's full screen by default. Performance for a smaller region of the screen (such as a specified window) is expected to be at least as good as that of the full screen. Only the capture size of 1024 x 768 was tested with 'Computer B' since it is an older computer with a small monitor, included for comparison purposes.

- screen content (*code*, *desktop*, *web*): For a realistic combination of text and images, tests were done capturing three different types of screen content: *code* in an IDE, the computer *desktop*, and a *web* page of an online newspaper (the screens captured in the 1280 x 1024 condition are shown in Figure 4.5).

4.3.3 Results

Tests measuring the speeds of shared screen image and remote pointing updates were done to examine the effect of performance on end-user experience. Results indicate that for the tile and remote pointing, updates are almost-instantaneous for both the owner and audience member. For the separate view, updates are almost-instantaneous for the owner, but there is some delay for the audience member, particularly if the screen is being viewed or shared from a slower network (e.g., from home). In other words, the Screen Sharing item's performance suffices for peripheral awareness, but could be improved for real-time discussion or collaboration between users involving the separate view.

Test Computer	Capture Area (pixels)	Time with Blur (seconds)		Time without Blur (seconds)	
		Mean	SD	Mean	SD
B	1024 x 768	0.37455	0.01338	0.14939	0.00825
A	1024 x 768	0.12968	0.00565	0.04861	0.00225
	1280 x 1024	0.24964	0.00659	0.07244	0.00285
	1200 x 1600	0.29531	0.00666	0.14641	0.00358
	1280 x 1024 1200 x 1600	0.62939	0.03315	0.29502	0.01368

Table 4.2 – Total time taken to update the shared screen image locally, with and without blurring.

Test Computer	Capture Area (pixels)	Time with Blur (seconds)		Time without Blur (seconds)	
		University	Residential	University	Residential
B	1024 x 768	0.43807	0.90900	0.21291	0.68384
A	1024 x 768	0.20378	0.75315	0.12271	0.67208
	1280 x 1024	0.31464	0.79655	0.13744	0.61935
	1200 x 1600	0.34124	0.68172	0.19234	0.53282
	1280 x 1024 1200 x 1600	0.66960	0.96767	0.33523	0.63330

Table 4.3 – Approximate total time taken to update the shared screen image in the tile for an audience member when both the owner and audience member are using the same network.

While running the tests, screen content did not appear to have too much of an effect on the local capture, blur, and resize times (see Section A.1 for details). Instead, screen content seemed to affect the size of the thumbnail and full image (because of the PNG compression algorithm used), thus affecting update times for audience members. For simplicity, screen content is disregarded as a factor in the remainder of this chapter, and average sizes measured from across the different screen contents (see Section A.2) are used in the following calculations.

4.3.3.1 Updating the Tile

Table 4.2 and Figure 4.6 show the total time taken to update the tile of the Screen Sharing item locally (for the owner) under the different conditions. Average times for the individual steps of capturing, blurring, and resizing the shared screen image are given in Tables A.1-A.3. As seen in Table 4.2, the average time without blurring is much faster than with blurring, suggesting that distortion should be applied to the thumbnail image

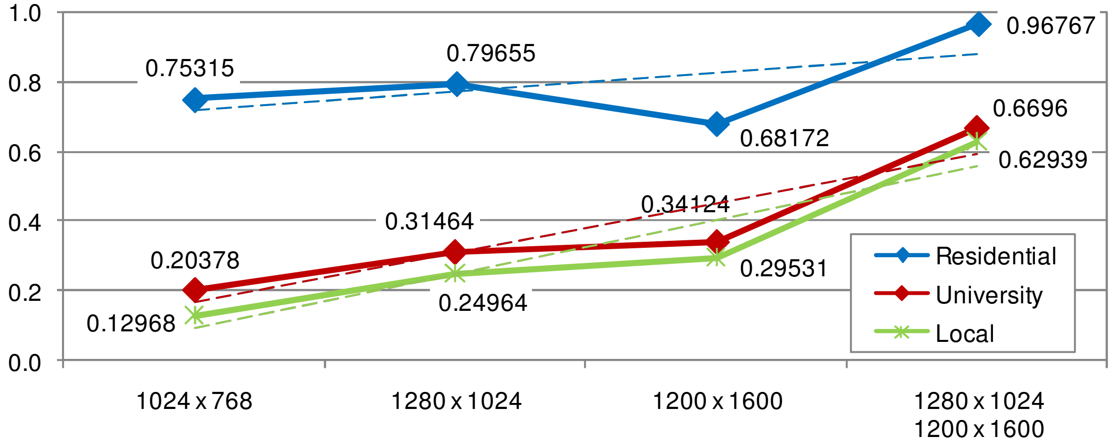


Figure 4.6 – Time in seconds to update a blurred, shared screen image tile for the owner (local update) and audience members (red for updates across a university network, blue for updates across a residential network). Note that the relationship between capture area and update time appears to be linear; see Section A.1 for discussion about the anomalous points.

rather than to the full image, especially when the full image is not needed. However, the average times both with and without blurring are still within the 1.0 second limit for keeping users' flows of thought uninterrupted, and the average time without blur is also below or close to the 0.1 second limit for having users believe that the system is reacting "instantaneously" (Nielsen, 1993) or the 10 frames per second minimum rate for having "convincing" motion in animation (Wagstaff, 1998).

Next, consider the update time of a tile for an audience member (Table 4.3 and Figure 4.6). As an example, suppose that the owner is using *Computer A* with a blurred capture region of 1280 x 1024, and that both the owner and the audience member are using the University of Calgary's network. The average thumbnail size would then be 23218 ± 4082 bytes (Table A.4). With an average upload speed of 5011 ± 2128 kbps and an average download speed of 6649 ± 2691 kbps (Table A.5), the transmission time for the thumbnail image from the owner's machine to the audience member's machine would be approximately 0.06500 seconds.

Considering the time to update the display is negligible compared to network transmission time, the total time to update the audience member's tile is then the transmission time added to the local update time. Thus, updating the shared screen in the tile for the audience member would take approximately 0.24964 + 0.06500 = 0.31464

Test Computer	Capture Area (pixels)	Time with Blur (seconds)		Time without Blur (seconds)	
		University	Residential	University	Residential
B	1024 x 768	1.68703	11.41732	1.46187	11.19216
A	1024 x 768	1.61718	12.64507	1.53611	12.56400
	1280 x 1024	2.03745	15.29170	1.86025	15.11450
	1200 x 1600	2.00366	14.66879	1.85476	14.51989
	1280 x 1024 1200 x 1600	3.57319	25.39756	3.23882	25.06319

Table 4.4 – Approximate total time taken to update the shared screen image in the separate view for an audience member when both the owner and audience member are using the same network.

seconds. If both the owner and audience member were using a slower network such as residential cable (see Section A.3 for network details), updating the tile for an audience member would then take approximately 0.24964 + 0.546914 = 0.79655 seconds.

While these times are above Nielsen's 0.1 second limit, most audience members will not perceive this lag unless they are conversing with the owner in real-time about artifacts seen in the shared screen, due to differences between what the owner may be talking about and what is visible on the audience member's view of the screen at the time. Yet, even then, the delay will be small.

4.3.3.2 Updating the Separate Window

For the owner, the time taken to update the shared screen image for the large view in the separate window will be similar to the time taken to update the tile (Table 4.2), since all distortions and resizing are done before any view is updated.

For an audience member, updating the shared screen image in the large view can take anywhere from one or two seconds to almost half a minute, depending on the network and the original capture area size (Table 4.4 and Figure 4.7). As an example, with the owner using 'Computer A' to capture a blurred region of size 1280 x 1024, it would take approximately 2.00366 seconds if both the owner and audience member are using a university network. For a residential network, an update would take approximately 14.66879 seconds.

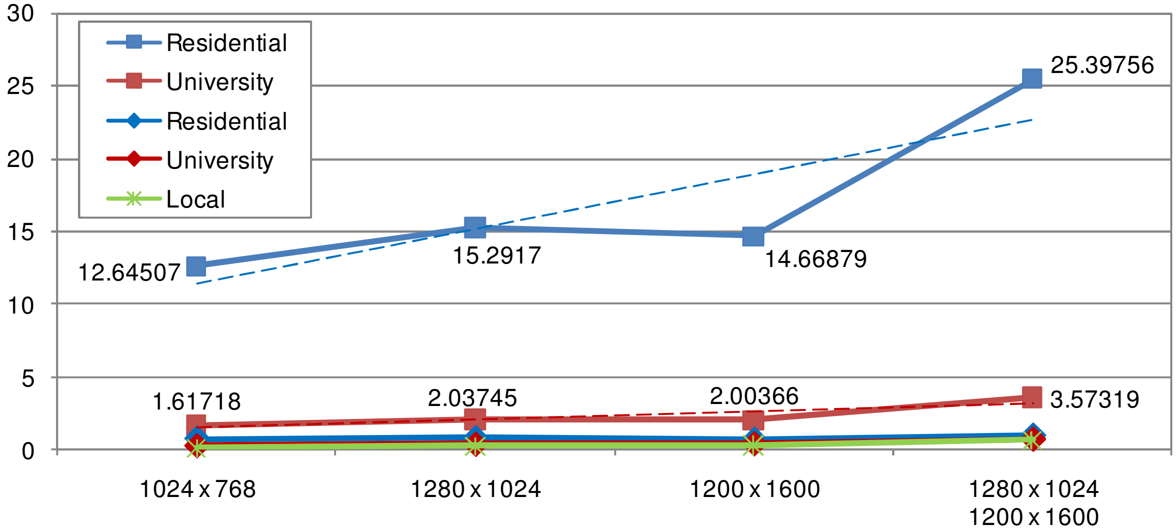


Figure 4.7 – Time in seconds to update a large, blurred, shared screen image for audience members (top red for updates across a university network, top blue for updates across a residential network). Times to update the tiles are also included as reference (bottom three lines; also shown in Figure 4.6). Note that the relationship between capture area and update time appears to be linear.

Note that the bottleneck in this process is the ‘slow’ residential upload speed; if an owner were sharing the full view of both screens (1280 x 1024 and 1200 x 1600) unblurred from the university and an audience member were to look at it from home, the total update time would decrease to 6.85577 seconds from the 25.06319 seconds it would take if both parties were at home.

Clearly, these delays will be noticeable by audience members, especially if they are waiting for a shared screen image to update before continuing with real-time discussion or collaboration over a shared artifact. However, if the separate view is being used for awareness, these delays would not be noticed by the audience member since the audience member would have no reference for when the screen was captured on the owner’s machine. Additionally, if an audience member were using the large view to analyze what an owner was working on, the fact that it is not completely ‘real-time’ would likely not matter, i.e., near-relative updates suffice for many purposes.

4.3.3.3 Using Remote Pointing

During remote pointing, only the coordinates of where the audience member is pointing at are sent across the network. These coordinates are stored in a structure that is 8 bytes large. When the average upload and download speeds from Table A.5 are used, the time it takes to transmit the coordinates when both users are on a university network is less than 0.0001 seconds. When both users are on a residential network, this time increases to approximately 0.000178. On DSL or cable connections, latencies of less than 100 milliseconds are typical. Thus, while the time to transmit coordinates may effectively increase to 0.100 seconds, the time taken for the pointer to move still seems almost instantaneous.

4.3.4 Opportunities for Future Work Relating to Performance

The results of the performance tests suggest several possible changes and additions to the Screen Sharing item for improving performance. In this section, I briefly discuss a few options that could noticeably improve end-user performance of the Screen Sharing item.

First, transmitting the large image across the network currently incurs a noticeable delay when the audience member is involved in real-time interaction with the owner using the separate window. Optimizations that have already been implemented for full screen sharing systems could also be implemented in the Screen Sharing item to improve the network transmission time. For example, the Screen Sharing item could only send pixel differences between consecutive shared screen images (Sarin, 1984), or use an adaptive image-compression scheme that changes depending on the network bandwidth and computer processing power (Frederick, 1994; Richardson et al., 1998).

Alternatively, the Screen Sharing item could take an approach of giving the ‘illusion’ that real-time updates occur, by showing an intermediate image while the large, full-quality image is being retrieved. For example, the Screen Sharing item could initially display a JPEG-compressed image while the PNG-compressed image is being transmitted (JPEG images are smaller in size but can cause text and lines in images to appear blurry, which is why it is not currently being used as the compression format for the Screen Sharing item). Or, the Screen Sharing item could initially display a medium-sized image while the large image is being retrieved. The thumbnail image could also be used as an intermediate image, but may be too small for real-time discussion about details of any shared artifacts seen in the shared screen. If this approach of using an intermediate image is used, the method of transitioning from the intermediate image to the final image should be carefully considered, as it may be distracting for the audience member if the images were constantly switching size and/or quality (e.g., as might happen if the owner is frequently updating their shared screen image).

Additionally, the time taken to update a thumbnail that includes distortion could be improved by resizing the image first, and then distorting it (rather than distorting the large image first, and then scaling it down to a thumbnail). Switching the two steps in this way may marginally increase the total amount of time spent on distortion if the large image is required (e.g., recall that the large image is only transmitted if someone is looking at the shared screen in the separate or remote pointing window) since both the thumbnail and the large image will have to be distorted. However, switching the two steps will make thumbnail updates much faster when the large image is not required.

For blur distortion in particular, the time required to blur a shared screen image can also be reduced by using a faster blurring algorithm. Deriche IIR (Deriche, 1990) is suggested by the developer of the Collabrary as one of several "computationally inexpensive filters [that] could be used instead [of t-he current implementation in the Collabrary] for real time video manipulation" (Boyle et al., 2000).

Finally, the Screen Sharing item currently updates shared screen images even when the owner's computer has been idle for some time. To reduce unnecessary network transmissions, the Screen Sharing item could automatically stop transmitting shared screen images once the computer has been idle for, say, five minutes (as the Screen Sharing item already detects computer idle time and turns a different colour to indicate that a computer has been idle for five minutes or more).

4.4 Summary

In this chapter, I showed how the Screen Sharing item was implemented as a Community Bar plug-in. The Screen Sharing item follows Community Bar's dMVC data distribution pattern while implementing a centralized, image-based approach to screen sharing.

When sharing an owner’s screen with other audience members, the Screen Sharing item first captures the region of the screen to share. Next, it applies any image distortions the owner uses as well as optimizes the shared screen image for transmission across the network. It then sends the shared screen image to the shared dictionary server, which notifies the audience members’ Screen Sharing items of the change. Finally, each of the notified audience members updates their view with the new shared screen image.

As a research prototype, the Screen Sharing item was implemented to be robust enough to deploy for use so that the effects of screen sharing for artifact awareness could be studied. Results from performance tests indicate that for the tile and remote pointing, updates are almost-instantaneous for both the owner and the audience member. For the separate view, updates are almost-instantaneous for the owner but there is some delay for audience members, particularly when the owner is sharing their screen from a network with a ‘slow’ upload speed. In other words, the Screen Sharing item’s performance suffices for peripheral awareness, but could be improved for real-time discussion or collaboration between users involving the separate view. Ways of improving performance could include sending only pixel differences across the network or using intermediate images to provide an ‘illusion’ of responsiveness to the end-user. For shared screens that are distorted to preserve privacy, re-ordering a couple of the steps in the process of sharing a screen could also improve performance.

Chapter 5. Evaluation

In the previous two chapters, I introduced the Screen Sharing item artifact awareness tool and explained the details behind how it was implemented. In this chapter, I describe the initial experiences and reactions that two different groups had while using the Screen Sharing item. First, I discuss findings from an initial user study done with members of our own research group. Then, I discuss findings from several interviews that were done with members of a commercial development team[1]. Finally, I highlight some of the major similarities and differences that were seen between the two groups' experiences and use of the Screen Sharing item.

5.1 Initial User Study

People inhabiting a common space naturally see the screens of their co-workers as they glance around, walk by, or are invited to take a closer look. However, using a shared screen in distributed groupware for artifact awareness is an unusual concept, and such a use will likely have to develop over time as part of a group's everyday practice.

Recruiting a group to use Community Bar (CB) and the Screen Sharing item over a long period of time is difficult for pragmatic reasons. Instead, I decided to introduce the Screen Sharing item to members of our own research group, who had already been using CB on a daily basis for over a year. While this was perhaps a biased group, there were several good reasons for using them:

[1] These interviews were done in partnership with Gregor McEwan, another Masters student in the Interactions Lab at the University of Calgary. Each interview consisted of two phases. Gregor conducted the first phase, which focused on CB; findings from this phase are briefly mentioned in (McEwan, 2006; Romero et al., 2007). I conducted the second phase, which focused specifically on the Screen Sharing item and will be reported on in this chapter. Analysis of collected data was done independently by each of us.

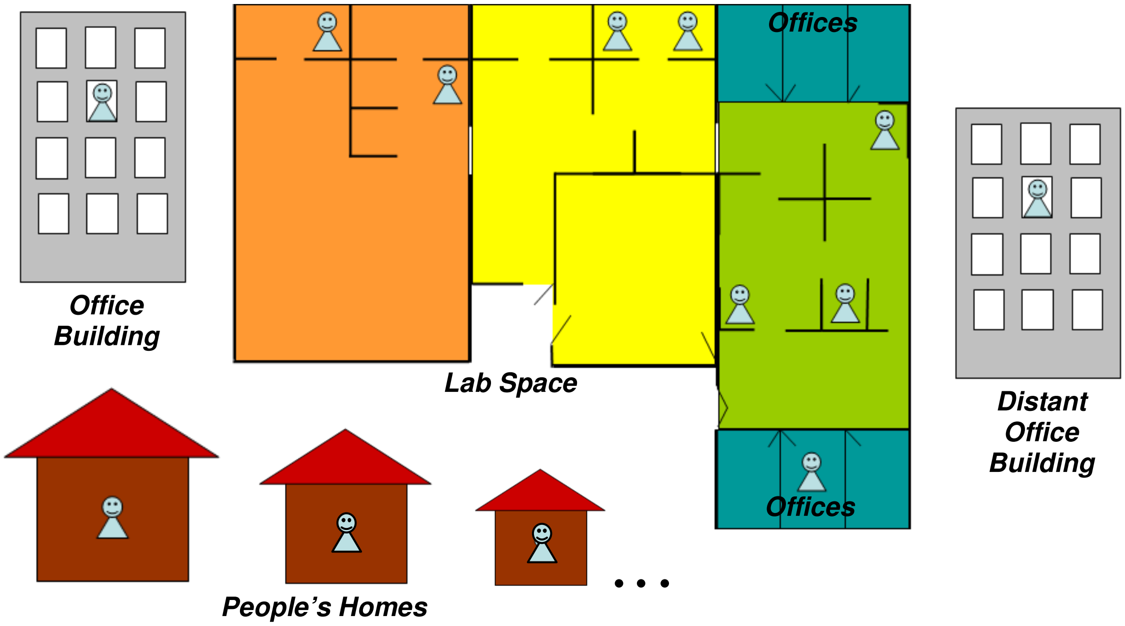


Figure 5.1 – Participants typically used CB from the lab or from home. One participant used CB from a work office in the same city, and another used CB from a work office in a distant city.

- members of the group had voluntarily used CB for their own use on a daily basis for over a year;
- as a group with an established culture of use of CB, they had already established a practice of balancing awareness provision with privacy for other media items;
- they were willing to aggressively use the Screen Sharing item for the initial deployment period and report on their usage;
- as most were knowledgeable about awareness systems, they could provide not only details of their personal use but reflective comments as well.

5.1.1 Participants

The group that was introduced to the Screen Sharing item included ten graduate students, faculty, research assistants, and former members of the research laboratory. People were both co-located and distributed. Most lab members primarily worked in one of three connected laboratory spaces, while faculty was located in separate offices. Figure 5.1 illustrates a map of the space, where icons of the people participating in the study are shown at their most typical seating location. This laboratory space was large enough that a person in one of the rooms would not normally see what a person in an adjoining room was doing. Group members were not always in the laboratory, as some tele-commuted when working at home (Figure 5.1, bottom left). More people tele-commuted in the evenings and on weekends. Former lab members also connected to CB from their distant work offices, one in the same city as the lab (Figure 5.1, left), one in a different city (Figure 5.1, right).

Most members of this group had a webcam and two monitors connected to their computers, and regularly used CB as a peripheral display on one of them. Members of this group were comfortable sharing information about themselves with others, and felt that they benefited from sharing.

5.1.2 Method

The Screen Sharing item was distributed to CB users in our research group as an installation package downloadable from a web page [grouplab.cpsc.ucalgary.ca/ cookbook]. When run, the installation package added the Screen Sharing item to CB. Group members were then free to use or not use the Screen Sharing item as they wished.

As this was not a formal study, no usage data was logged by the system. Instead, people were asked to email in descriptions of use and other comments as they used the system over a two week period, and several people were interviewed in depth to discuss details. As a CB user, I remained logged into CB much of the time and collected notes whenever I saw people discussing shared artifacts or desktops in CB. The key experiences and reactions people had to the Screen Sharing item, based on both these user reports and my direct observations, are summarized in the next section.

5.1.3 Initial Experiences and Reactions

With this initial user study, the primary goal was to observe how people shared their screens and interacted with the screens being shared by others. Based on discussions with the group using CB as well as observations made during the deployment period, initial experiences show that people use the screen-sharing awareness tool for several purposes:

to maintain awareness of what others are doing, to monitor progress and coordinate joint tasks, to help determine when another person can be interrupted, to project a certain image of themselves, and to engage in serendipitous conversation and collaboration. Also, people used the privacy-protection strategies built into the system to balance awareness and privacy. In the following sections, I describe in more detail the key experiences and reactions people had to the Screen Sharing item.

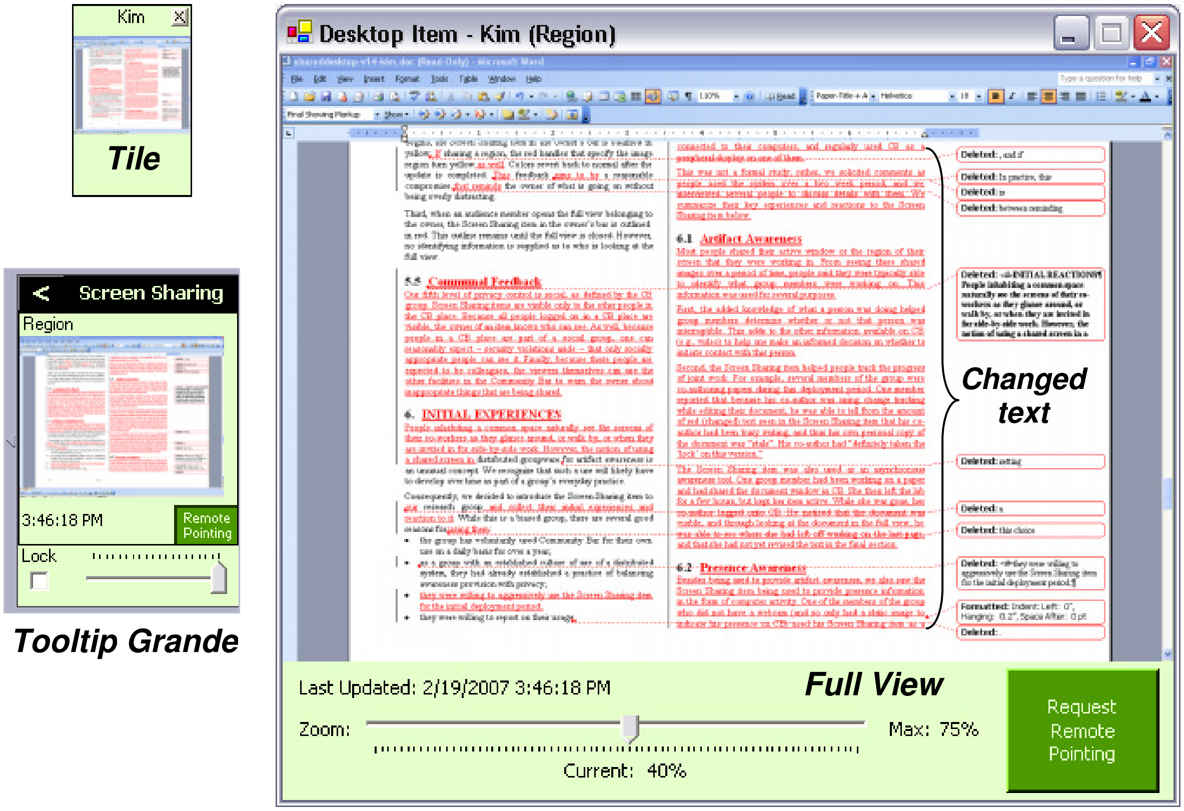
5.1.3.1 Artifact Awareness

Most people shared their active window or the region of their screen that they were working in. From seeing these shared images over a period of time, people said they were typically able to identify what group members were working on. This information was used for several purposes.

First, the added knowledge of what a person was doing helped group members determine whether or not that person was interruptible. This adds to the other information available on CB (e.g., video) to help people make an informed decision on whether to initiate contact with another person.

Second, the Screen Sharing item helped people track the progress of joint work. For example, several members of the group were co-authoring papers during this deployment period. One member reported that because his co-author was using ‘change tracking’ while editing their document, he was able to tell from the amount of red (changed) text seen in the Screen Sharing item that his co-author had been busy writing, and thus his own personal copy of the document was “stale” (Figure 5.2). His co-author had “definitely taken the ‘lock’ on this version.”

The Screen Sharing item was also used as an asynchronous awareness tool. One group member had been working on a paper and had shared the document window in CB. She then left the lab for a few hours, but kept her item active. While she was gone, her co-author logged onto CB. He noticed that the document was visible, and through looking at the document in the full view, he was able to see where she had left off working on the last page, and that she had not yet revised the text in the final section.



Last Updated: 2/19/2007 3:46:18 PM

Figure 5.2 – Using screen sharing to track progress on a paper.

5.1.3.2 Presence Awareness

Besides being used to provide artifact awareness, the Screen Sharing item was also used to provide presence information in the form of computer activity. One of the members of the group who did not have a webcam (and so only had a static image to indicate his presence on CB) used his Screen Sharing item as a replacement for his Presence item. Since he used a frequent auto-update rate for sharing his screen, it was easy to see when he was at his computer because windows would be scrolled up and down or be moved about. It was also possible to tell when he had been away from his computer for some time, as the Screen Sharing item's background colour automatically darkened when the computer had been idle for five minutes or more.

5.1.3.3 Opportunistic Interactions

Conversations would sometimes arise as a consequence of people seeing artifacts in the Screen Sharing item. For example, one member saw his co-author working on their paper, and asked how it was going. His co-author responded “It’s going ok – I’ve got some inspiration about how to proceed for a bit.” They then proceeded to coordinate when each would work on it, deciding that the co-author would continue writing for the day, and then pass the draft on to the other author. In another instance, one member of the group saw some interesting-looking design images on another’s desktop. When asked what they were for, she was told that they were t-shirt designs. This led to a brief conversation about that individual’s extra-curricular activities outside the lab, which were not widely known before.

These serendipitous conversations would occasionally transition into remote pointing sessions, which were typically used to discuss joint work between two people. Most of these sessions occurred when at least one of the participants was working from home for the day. For example, a group member noticed that his co-author was working on a figure for their paper after he had sent her an email with some suggestions for improving it. They used remote pointing to discuss which parts of the figure should be changed. Afterwards, the group member was able to peripherally see his co-author making refinements to the figure, and she would intermittently ask him to check his view of her shared screen in order to get feedback on the image. In another example, a group member was working on an initial draft of a paper that her co-author hadn’t seen yet. Her co-author noticed the document in the Screen Sharing item, but the image was blurred, so he asked her to unblur it. They then went into a brief remote pointing session to discuss the title and abstract.

5.1.3.4 Focused Collaboration

People also reported using the Screen Sharing item for focused interactions after they had already begun a conversation or a meeting. For example, two co-authors were discussing a paper via VoIP and had to look at an image. They started a remote pointing session and used the telepointer to make sure they were talking about the same parts of the image. In another instance, one group member had asked for some visualizations from another member on CB. She wasn’t sure exactly which ones he wanted, so she shared them on her screen in order to confirm with him that they were the correct ones before she sent

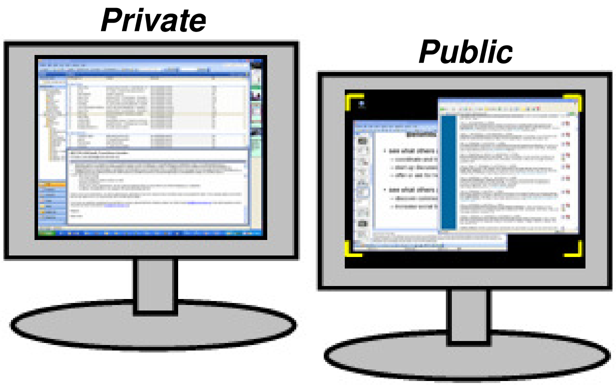


Figure 5.3 – Separating private and public information onto different displays.

them. In a third case, one group member was helping another work on her poster. They used the Screen Sharing item and remote pointing to try out and discuss different layouts.

5.1.3.5 Privacy Issues

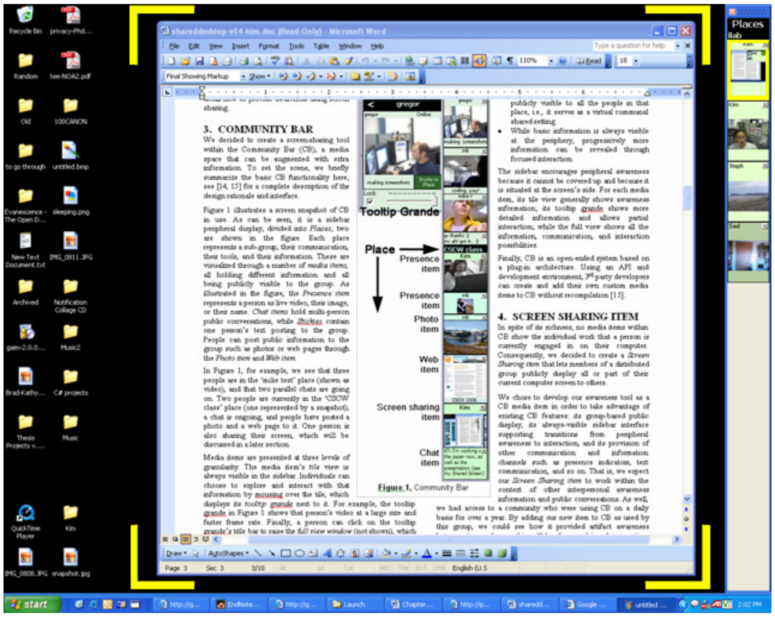
People adopted different strategies to protect their privacy while sharing their screens.

First, because most members of the group had two monitors, some chose the strategy of separating semi-public information (i.e., information that they were willing to share) and private information (i.e., information that they preferred not to share) onto different displays (Figure 5.3). Some chose to share their entire public screen, while others chose to share only a specific region of it. For example, one group member reported “I have a two screen system, where I normally read email on the right screen, and do work on the left. I decided that I am happy to share my work (left) screen, so I set the region to the top half of that (using the idea that things above the fold are more relevant).” This strategy of separating public and private information onto different displays is one example of the ‘partitioning’ that Grudin (2001) observed when studying how people use multiple monitors.

One person questioned this public / private separation as it differed from real-life activities. He liked having a screen where things weren’t publicly visible, but he wasn’t sure why, “because anyone can walk by [in the lab] and see [it]”. This perception of digital artifacts as being private when in fact they are semi-public is partly a result of current workplace etiquette, which suggests that people should refrain from looking closely at another’s display unless invited to do so. In CB, the act of posting a Screen Sharing item acts as a tacit invitation to look closer if interested, and so people may have the feeling that others are looking closely at what is being shared on their screen regardless of whether anyone actually is. In this regard, the feedback from the Screen Sharing item that indicated to people that someone else was looking at their desktop using the full view was only partially effective. While useful for making people more conscious of what they were sharing, people also wanted to be able to identify who was looking at their desktop without that person having to explicitly tell them. This feedback also unintentionally discouraged people from looking at others’ shared screens in the full view; one group member reported that there were times when he had wanted to look at another’s desktop using the full view, but was slightly hesitant to do so because the other person would then know that someone was looking. This suggests that additional information should be supplied, which I will discuss in Section 7.3.1.

Second, some people chose to blur the screen image they shared so that text in windows would not be legible to others (Figure 3.9a). In fact, almost everyone who chose to share their active window opted to blur it. This is because unlike the private / public separation strategy for screen sharing mentioned above, sharing the active window is indiscriminate in what it displays. Even in this case though, people didn’t have a problem with others being able to see the basic tasks they were working on. Rather, they were not always comfortable with sharing the details. This was particularly true of activities involving personal communication, such as checking email or instant messaging chats. People who had to work with confidential information such as study data (protected by ethics reporting) or product source code (e.g., people working offsite in industry) also blurred their shared screen images to obscure the details, but still felt comfortable giving others an idea of what they were doing.

Third, people generally limited the amount others could zoom into their full view to less than the original resolution (Figure 3.9). When people moved into interaction, such as during discussions about shared artifacts visible in the full view or during remote



shareddesktop-v14-kim.doc (Read-Only) - Microsoft Word

sharing.

3. COMMUNITY BAR

We decided to create a screen-sharing tool within the Community Bar (CB), a media space that can be augmented with extra information. To set the scene, we briefly summarize the basic CB functionality here, see [14, 15] for a complete description of the design rationale and interface.

Figure 1 illustrates a screen snapshot of CB in use. As can be seen, it is a sidebar peripheral display, divided into *Places*; two are shown in the figure. Each place represents a sub-group, their communication, their tools, and their information. These are visualized through a number of *media items*, all holding different information and all being publicly visible to the group. As illustrated in the figure, the *Presence item* represents a person as live video, their image, or their name. *Chat items* hold multi-person public conversations, while *Stickies* contain one person's text posting to the group. People can post public information to the group such as photos or web pages through the *Photo item* and *Web item*.

In Figure 1, for example, we see that three people are in the 'mike test' place (shown as video), and that two parallel chats are going on. Two people are currently in the 'CSCW class' place (one represented by a snapshot), a chat is ongoing, and people have posted a photo and a web page to it. One person is also sharing their screen, which will be discussed in a later section.

Media items are presented at three levels of granularity. The media item's *tile view* is always visible in the sidebar. Individuals can choose to explore and interact with that information by mousing over the tile, which displays its *tooltip grande* next to it. For example, the tooltip grande in Figure 1 shows that person's video at a large size and faster frame rate. Finally, a person can click on the tooltip grande's title bar to raise the *full view window* (not shown), which

Figure 1. Community Bar

publicly visible to all the people in that place, i.e., it serves as a virtual communal shared setting.

- While basic information is always visible at the periphery, progressively more information can be revealed through focused interaction.

The sidebar encourages peripheral awareness because it cannot be covered up and because it is situated at the screen's side. For each media item, its tile view generally shows awareness information; its tooltip grande shows more detailed information and allows partial interaction; while the full view shows all the information, communication, and interaction possibilities.

Finally, CB is an open-ended system based on a plug-in architecture. Using an API and development environment, 3rd party developers can create and add their own custom media items to CB without recompilation [15].

4. SCREEN SHARING ITEM

In spite of its richness, no media items within CB show the individual work that a person is currently engaged in on their computer. Consequently, we decided to create a *Screen Sharing item* that lets members of a distributed group publicly display all or part of their current computer screen to others.

We chose to develop our awareness tool as a CB media item in order to take advantage of existing CB features: its group-based public display, its always-visible sidebar interface supporting transitions from peripheral awareness to interaction, and its provision of other communication and information channels such as presence indicators, text communication, and so on. That is, we expect our *Screen Sharing item* to work within the context of other interpersonal awareness information and public conversations. As well, we had access to a community who were using CB on a daily basis for over a year. By adding our new item to CB as used by this group, we could see how it provided artifact awareness

Figure 5.4 – Auto-update warning. The owner's tile is outlined in yellow in the bar and the handles also turn yellow (enhanced here) five seconds before the screen region is captured.

pointing sessions, people would increase the maximum zoom. After these discussions, they would then decrease it.

Fourth, people reported that the auto-update feedback, where the handles and tile changed colour to indicate an impending update (Figure 5.4), was particularly effective as it served as a constant reminder that the region was being shared. There were several cases reported where private windows were almost shared when they should not have been. For example, one faculty member began setting examination questions on his public screen, where he normally did his work. Fortunately, the visible warning from the Screen Sharing item that an update was about to occur reminded him that the exam questions should not be publicly available. He then moved the document from the shared region to the private secondary screen before the exam questions could be seen by others.

There was some concern from audience members that people could see too much of others' desktops; after observing on CB that one person was composing email, and that another was reading sports news online, one group member commented "So here I am perusing people's desktops [in the full view]... Hmm, am I seeing too much?" This concern was surprising; we expected that people sharing their desktops would be concerned about sharing too much information, but we did not expect that audience members would feel uncomfortable seeing too much of someone else's desktop. This idea of 'reveal' (Boyle and Greenberg, 2005a) can actually heighten privacy, as it allows one person to warn others when they are unintentionally revealing something. For example, one person noticed that a colleague working at a distant industrial site was working on code development, where full details were visible. He used CB to start a discussion with that person, where he asked if there would be concerns about proprietary code being revealed outside that site. He then taught the person how to use blurring, where levels could be set to reveal coding activity without revealing contents.

Even with the privacy controls and feedback, there were some members of the lab who were not comfortable sharing their desktop using the Screen Sharing item. One member of the lab was concerned that if others did not see him working on his computer, they would think that he was "slacking off". In contrast, another member of the group who used the Screen Sharing item commented that one reason he liked it was because it "lets me project a certain image of myself. I can use it to indicate I'm working, or pretend that I'm working". These incidents are examples of some of the privacy maintenance issues discussed by Voida et al. (2005).

5.1.3.6 Distraction Issues

When many desktops were being shared on CB, people found it difficult to find the ones they wanted to see. Most people were only interested in a subset of the desktops being shared, such as task-oriented subsets that included only the desktops of people working on different aspects of a collaborative task (e.g., paper writing), or social subsets that included the desktops of close friends. This could be easily resolved using the Place feature in the Community Bar to create a more focused sub-group, but this practice had not yet been established by this larger group (Romero et al., 2007).

Contrary to what I expected, no one said that they found the Screen Sharing item or its auto-update warnings (Figure 5.4) distracting. In fact, there were cases when an artifact on someone's 'unimportant' screen caught the eye of another person. This sometimes resulted in a purely serendipitous and opportunistic conversation, which is one of the benefits of having universal awareness of things that are not initially of interest.

5.1.4 Summary of Use by the Internal Research Group

Based on my discussions with the group using CB, the majority of uses of the Screen Sharing item were to maintain awareness of what others were doing and to glance at details of this activity through either the tooltip grande or the full view. This sometimes led to brief interactions outside of the Screen Sharing item, e.g., when people would use the Chat item to discuss things they could see. For work-oriented activities, these chats were often part of a broader longitudinal discussion of how the joint work was progressing as a whole. Discussions only sometimes proceeded to remote pointing, usually when focused interaction over the artifact was required. Full screen-sharing functionality (e.g., application sharing) was not frequently requested.

The people who found the Screen Sharing item most useful were those working together on a joint paper or project. They were able to see which parts their collaborators were working on, and were able to discuss various things about the paper or project, sometimes using remote pointing. Those who weren't directly collaborating with other CB users mainly found the Screen Sharing item useful for determining whether or not someone could be interrupted at a particular point in time.

People adopted different strategies to protect their privacy while sharing their screens. The most common strategy was to separate public and private information onto different displays. While this generally worked, 'accidents' still happened sometimes, such as when people forgot that a particular screen was being shared.

Design ideas and reflections from these observations will be discussed in Section 7.3. For now, these results point to a broader use of screen sharing, ranging from artifact awareness, to monitoring activities, to brief discussions, and sometimes to focused work.

5.2 Second Evaluation

Two months after the Screen Sharing item was deployed to our research group, it came to our attention that members of a commercial development team were also using CB and the Screen Sharing item, which had been introduced to them by a former member of the lab who had gone to work at the company. In this section, I present findings from initial interviews done with several of the people in this industry group. This was again not a formal study; rather, we took the opportunity to get feedback from an outside group of users when the opportunity arose. In particular, we were interested in finding out whether this industry group's use of CB and the Screen Sharing item differed greatly from our own research group's use.

5.2.1 Method

Semi-structured interviews were conducted on-site at the company. Each interview consisted of two phases, and only one interview lasted for more than half an hour. The first phase focused on CB and the group's usage of it, as well as background information about the group. The second phase focused specifically on the Screen Sharing item. The questions covered in this second phase of the interview can be found in Section B.3, and I discuss the findings from this stage in the next part of this chapter.

5.2.2 Participants

The development team consisted of seventeen people located in the same building, though they were scattered across the floor and some members were on different floors. Of that group, an estimated (by participants) ten to fifteen people had tried CB, though the core group of daily users was from five to seven people. From this core group of users, we interviewed four volunteers. Three of them were developers, each from a different sub-group, and one was an internal user experience consultant. The internal user experience consultant had been part of our lab several years ago and had used CB before, but was not otherwise involved in the CB project other than as an end-user.

Another core CB user, not included in the interviews, was also a former member of our lab, and was the one who suggested to the group that they try CB. Though he was also not directly connected to the CB project, he had been a heavy CB user in the lab (which he had more recently left) and he played a large role in encouraging the development team to try out CB. In addition to CB, the group also planned to try out another ICQ client; according to the manager, the group was actively looking for a tool to support awareness and casual communication among team members.

At the time of the interviews, the group had been using CB for about three weeks, and all our participants had been using CB for at least two weeks. Two of our four participants shared their screens regularly, one shared his screen occasionally, and the fourth had tried it out once or twice but did not currently share his screen (because he felt that he didn't have anything to share at the moment). While most members of the group had two monitors, only one person had a webcam. The four participants that we interviewed only connected to CB while they were in the office. Their key experiences and reactions to the Screen Sharing item are summarized in the next section.

5.2.3 Their Initial Experiences and Reactions

With these interviews, the primary goal was to investigate whether an industry group's use of the Screen Sharing item greatly differed from our own research group's use. While we were only able to interview four of the CB users on the commercial development team, it appears that their experiences with the Screen Sharing item were similar to the experiences of our in-house group. People in the industry group also used the Screen Sharing item to maintain awareness of what others were doing, help determine others' availability, project a certain image of themselves, and engage in opportunistic conversation and collaboration.

Yet, there were also some differences in how the two groups used the Screen Sharing item. Interestingly, the company maintained a Bridgit [www.smarttech.com] server that anyone could connect to at any time for desktop conferencing or remote collaboration. In addition to full screen sharing, Bridgit offers a number of additional features such as remote highlighting and annotation, built-in VoIP, and support for multiple webcams. While Bridgit has some features similar to CB, it is intended to be used for focused collaboration, not awareness or casual interaction. Having Bridgit freely available may have contributed to some of the differences seen in usage of the Screen Sharing item by the two groups, such as how in the industry group, people did not seem to opportunistically track or coordinate joint work that much using the Screen Sharing item. There also seemed to be less focused collaboration occurring through the Screen Sharing item, though a number of opportunistic interactions were still reported. People in the industry group also seemed to share their screens less than the people in our own research group did, though privacy did not appear to be a large concern for them.

In the following sections, I describe in more detail the key experiences and reactions that people in the industry group had to the Screen Sharing item. I highlight and discuss the similarities and differences in how the industry group used the Screen Sharing item vs. how our own research group used it in Section 5.3.

5.2.3.1 Artifact Awareness

People in the industry group typically shared their full screen or a region of their screen. They usually shared what they were working on, though one participant said that he often focused in on one tiny region of his screen such as his MSN Messenger display picture to share, as a joke. Though an unintended use of the system, it still sometimes led to conversation or banter about what he had shared.

Participants said that they could identify what they saw on others' desktops "surprisingly really well". All reported being able to identify activities such as coding and web surfing, and were able to recognize some programs that others were using just from seeing the outlines of the windows (Figure 3.4). Again, the added knowledge of what a person was doing helped group members determine whether or not that person was available for interruption. Having this availability information was the most common reason participants said that they liked seeing others' screens.

In contrast to our research group's experience with the Screen Sharing item, the participants from the industry group did not seem to use it to opportunistically coordinate or track progress of each others' work. This may have been because our participants were from different sub-groups of the project and so did not typically work closely together. Also, during the time that the group had been using CB, they were in-between projects and thus felt “no real big push” to get one particular thing done or to act together towards any one particular goal. Rather, they were mainly doing bug fixes and maintenance.

Because participants were from different sub-groups, they didn’t usually hear about each others’ projects. In combination with being able to overhear conversations that occurred in CB, being able to see what others were working on helped give people a better idea of what was going on with others’ projects. This in turn led at least one participant to feel like he knew the other team members a bit better.

5.2.3.2 Presence Awareness

Because none of the participants interviewed (and only one person in the whole group) had a webcam, the Screen Sharing item was found useful for presence information within the commercial development group more so than within our own research group. The Screen Sharing item acted as a partial replacement for a webcam by showing when people were using their computers. It was only a partial replacement in terms of providing presence awareness because the Screen Sharing item might still show someone as ‘away’ when they were in fact present but not using their computer (e.g., they were reading at their desk).

However, the Screen Sharing item did indicate useful information about another’s presence and availability that would not have been captured by a webcam. For example, a couple of the people who shared their screens on CB also had Macintosh computers that they regularly connected remotely to, e.g., through VNC (Richardson et al., 1998). Because they were sharing their screens, their Mac screens would be captured by the Screen Sharing item when they were connected, indicating to others on CB that email and IM messages to their PC would likely go unnoticed until they disconnected. In other words, although they might still be available to people walking by or calling in, electronic messages would probably be unanswered while they were connected to their Mac.

5.2.3.3 Opportunistic Interactions

In the industry group, opportunistic interactions often seemed to be initiated by the person sharing their screen telling another on CB to go look at a shared artifact or region.

This was in contrast to our research group’s experience, where people tended to notice and ask about shared artifacts without additional encouragement from the person sharing. For example, in the industry group, one group member told a participant to check out a blog entry on his shared screen, and they moved into conversation and remote pointing about it. Another participant used the Screen Sharing item to opportunistically show a team member a bug he had found in what they were working on together. That same participant commented that there was another person that he was working closely with that was not on CB, but he wished that the person was, so that they could share their screens with each other.

Few conversations seemed to transition into remote pointing; participants reported using remote pointing mainly to try it out.

5.2.3.4 Focused Collaboration

Members of this industry group did not use the Screen Sharing item for focused interactions much, partly because the company maintained a Bridgit server that anyone could connect to at any time. Though there was some overhead for creating or joining a desktop conferencing session, Bridgit allowed full remote control of another’s desktop as well as had integrated VoIP. In fact, two participants said that they would like to have a link to Bridgit from the Screen Sharing item. A similar link was implemented for the Notification Collage Desktop media item (Rounding, 2004) and it would be nice to have in the Screen Sharing item in order to provide a more complete transition from awareness to full groupware.

5.2.3.5 Privacy Issues

People in the industry group used somewhat similar privacy-protecting strategies as the people in our own research group. In particular, most members of the industry group also had two monitors and reported separating their public and private information onto different regions of their displays. Having two monitors seemed to alleviate many of the privacy concerns that people had; one participant commented, “I can always open up an IDE and make them think that I’m working on something. That’s good for me because I have two monitors. If people have [only] one [monitor], probably [people might feel differently about screen sharing]."

Unlike the participants from our research group, none of the participants from the industry group reported regularly sharing a specific window or their currently active window. Also, none regularly distorted their shared screen image. One person typically shared his screen at the maximum (original) resolution, and another shared his screen at the original resolution about half the time. One participant mentioned that he would like a way of blocking out certain regions of his screen so that they couldn't be seen by others.

In most cases however, privacy did not seem to be a large concern for these participants. None reported having any concerns about screen sharing either before or after trying the Screen Sharing item. This may partly have been because of the current environment or group they worked with; when asked about whether he had any concerns about sharing his screen, one participant responded, "here [at this company], not really. I guess I'd never thought about… if I switch away from my paper because I'm falling asleep and go check my personal email… people might be able to read it, but then again, what do I get in my personal email that's really all that private anyway." Another participant mentioned that while he only wanted to share his screen at certain times (he didn't like people "over [his] shoulder all the time"), as long as he could get his privacy when he needed it, he had no real concern with using the Screen Sharing item.

5.2.3.6 Distraction Issues

Again, no one seemed to find the Screen Sharing item or its auto-update warnings distracting. One participant did shrink others' Screen Sharing items when he wasn't interested in what they were doing (he was only interested in what they were doing when he wanted to talk to them).

People in this group seemed to be more careful about trying not to distract others and trying not to clutter up the bar, perhaps because they were working in an industry environment where productivity was very important. Two participants mentioned that they only shared their screens when they had something to share, with one commenting that the reason he didn’t share his screen all the time was because doing so takes up extra space in the bar. There is a trade-off between awareness and distraction; while sharing screens all the time can lead to opportunistic interactions triggered by shared artifacts, a cluttered bar can make it difficult to find shared screens and artifacts of interest. These issues may be specific to CB and the way CB displays information; alternative designs for screen-sharing awareness tools that might address the concerns these group members had about ‘frivolously’ taking up others’ screen space are discussed in Section 7.3.3.

5.2.4 Summary of Use by External Industry Group

Based on the interviews with members of this industry group using CB, the majority of uses of the Screen Sharing item were to check others’ availability and to show others things such as artifacts or actions on one’s screen. People liked not having to walk over to others’ desks or have others come over; one participant felt that the primary benefit of using the Screen Sharing item was that he could easily show people things on his screen without having them come over to join him in person.

Again, the people who found the Screen Sharing item most useful were those working together on something. Unfortunately, our participants were from different sub-groups of a project and so did not always work closely together. Also, during the time that the group had been using CB before the interviews, they were in-between projects and thus did not feel the need to work together towards any one particular goal. Still, some opportunistic interactions did occur, though they were often more social in nature.

Few conversations transitioned into remote pointing or focused collaboration, though full screen-sharing functionality was more frequently requested from this industry group than from our own lab group. This may have been partly because Bridgit was available to the industry group for desktop conferencing or remote collaboration whenever they wished to use it. Still, the Screen Sharing item was found to be useful for awareness and for creating opportunistic interactions. It was also found valuable for determining others’ availability and sharing artifacts and computer activities with others.

5.3 Discussion

The research group and the industry group used the Screen Sharing item for many purposes. In this section, I highlight the major similarities and differences in how the two groups experienced and used screen sharing for awareness.

I was only able to interview four participants from the industry group, rather than observe them using the Screen Sharing item over a period of time like I did with the ten participants from the research group. Consequently, some of the similarities and differences in usage that I saw between the two groups may not be as large as they appear; more in-depth studies need to be done with these or other groups to refine these findings.

5.3.1 Two Groups

First, I discuss the similarities and differences between the two groups.

Both the research group and the industry group were composed of people involved in research or development relating to computer science. During the deployment periods, the majority of people in the research group were co-authoring papers (working closely with one or two others) and the majority of people in the industry group were (individually) working on bug fixes and product maintenance. People in the industry group had free access to a Bridgit server that their company maintained for them; typical uses of Bridgit included desktop conferencing and remote collaboration.

Members of both groups typically had two monitors each. While many members of the research group had a webcam, only one member of the industry group did. Considering that every participant in Romero et al.'s study (2007) reported that their primary motivation for using CB was because of the rich awareness they gained from being able to see webcam snapshots of people, the industry group likely had somewhat of a different experience using CB than the research group did. The research group had also been using CB for over a year, whereas the industry group had only been using CB for several weeks. Still, at the time of each study, the two groups had been using the Screen Sharing item for about the same amount of time.

5.3.2 Similarities

The following are the major similarities seen between how the two groups experienced and used screen sharing for awareness.

Similarity: Most useful for particular subgroups

Screen sharing seemed to be most useful for people working closely together on a joint task or project. By being able to see each others' screens, people were able to opportunistically coordinate as well as track progress of their joint work. They were also able to easily check whether collaborators were interruptible for work-related conversation. Social subgroups of good friends also found the Screen Sharing item particularly useful; people would commonly use it to check if others were busy working, or if they were free to take a break.

Similarity: Valuable for lightweight casual interaction

When people used the Screen Sharing item to check whether others were available, they particularly liked being able to check without having to actually walk over to that person's desk. People also liked being able to share artifacts and activities with others without having to walk over or have others come over. Using the Screen Sharing item, it was easy for people to tell others to look at their shared screen on CB and then move into conversation about what was being shared. For remote people, the Screen Sharing item enabled them to easily check availability or share artifacts when they normally wouldn't have been able to do so.

Similarity: Privacy not that large a concern

People from both groups had fewer privacy concerns than I expected. Having two monitors seemed to help alleviate much of people's concerns about sharing information since they could then separate public and private information onto different displays, sharing only one. Also, with more screen space, it was possible to 'pretend' to be working in the shared region while actually doing other activities in the non-shared regions. People in both groups recognized that they could project this image of themselves as being productive and thus were not too concerned about sharing their screen. The privacy controls in the Screen Sharing item also helped people balance awareness and privacy (though people from the two groups used the privacy controls in different ways). Further analysis of how the Screen Sharing item affects people's privacy is given in the next chapter.

5.3.3 Differences

The following are the major differences seen between how the two groups experienced and used screen sharing for awareness.

Difference: How opportunistic interactions were initiated

Surprisingly, opportunistic interactions in the industry group often seemed to be initiated by the person sharing their screen telling another on CB to go look at a shared artifact or region. In contrast, people in the research group tended to notice and ask about shared artifacts without prompting from the person sharing their screen. This may have been because people in the research group generally shared their screens all the time, whereas people in the industry group tended to share their screens only when they felt that they had something to share.

Difference: How often people shared their screen

People in the industry group seemed to be more conscious about only sharing their screens when they felt that they had something to share. One reason given was that sharing a screen takes up extra space in the bar. This desire to avoid cluttering up the bar was a little surprising considering that there were about half the number of CB users in the industry group than in the research group. Also, from the interviews, it did not sound like the industry group shared an excessive amount of websites or photos that would take up a lot of space in the bar. Still, it appears that this carefulness to not take up extra space in the bar partly contributed to why the industry group shared their screens less than the participants in the research group did.

Difference: How often people collaborated together

There was also less focused collaboration through the Screen Sharing item reported in the industry group than in the research group, as well as fewer opportunistic interactions involving coordination or tracking progress of joint work. This may have been a result of the industry group members not working closely together on joint projects, and consequently not needing to do much collaborative work or coordination together overall. Alternatively, the availability of Bridgit (with its enhanced screen sharing features) may have made it easier for the industry group members to use that rather than the Screen Sharing item when they needed to check progress or do focused collaboration.

Despite not using the Screen Sharing item much for focused collaboration, people in the industry group still found the Screen Sharing item useful. This shows that there is a difference between shared screens for awareness and shared screens for focused collaboration, and highlights the importance of screen sharing for artifact awareness, something not previously promoted or discussed in the product or research literature.

5.4 Summary

Screen sharing was originally created to give collaborators the ability to do focused joint work across distances (Engelbart and English, 1968). The initial experiences people had reveal that this was one of the ways in which the CB Screen Sharing item was used. Yet, these experiences also reveal the importance of screen sharing for artifact awareness.

Based on discussions with the two groups using CB, the majority of uses of the Screen Sharing item were to maintain awareness of what others were doing and to opportunistically share computer artifacts or activities with others. Screen sharing for awareness was found particularly useful for certain subsets of the two groups, such as people working closely together and people who were good friends. The Screen Sharing item also helped people engage in lightweight casual interaction by allowing them to check availability or share artifacts onscreen without having to leave their desk. Privacy was less of a concern than expected, with most people having no problem sharing regions of their screen, or using the various privacy controls in the Screen Sharing item to balance awareness with privacy.

It remains to be seen whether other groups adopt the Screen Sharing item in a similar way as these two groups did and what happens over long periods of use. More in-depth studies need to be done with other groups to formally observe how they adopt the screen-sharing awareness tool for their own use. For now, these results point to a broader use of screen sharing, ranging from artifact awareness, to determining availability, to opportunistic discussions, and sometimes to focused work.

In the next chapter, I present a theoretical analysis of the effects the Screen Sharing item might have on its users' privacy.

Chapter 6. Theoretical Analysis of Privacy

In the preceding chapter, I reported on initial uses of the Screen Sharing item by two groups. As seen, privacy was less of a concern than expected, with most people using the various controls in the Screen Sharing item to balance privacy and awareness. However, these evaluations were with limited user communities (somewhat similar to each other), and thus their use may not predict other communities' concerns about privacy.

In this chapter, I apply Boyle's privacy framework and method of privacy analysis (Boyle, 2005) to the Screen Sharing item in a further exploration of how the Screen Sharing item can affect its users' privacy. As part of this process, I articulate possible privacy-related problems for potential users and examine the assumptions made during the design of the Screen Sharing item. I then relate findings from this theoretical analysis to what was seen in the informal evaluations, and briefly discuss how use of the Screen Sharing item might differ for other groups. I conclude by outlining potential areas of future work relating to privacy revealed by identifying privacy-related omissions in the current version of the Screen Sharing item.

In the following sections, vocabulary terms from Boyle's privacy framework are italicized when first used. Brief definitions of these vocabulary terms are given, but the reader is referred to the full definitions found in (Boyle, 2005) and (Boyle and Greenberg, 2005a) for greater detail.

6.1 Applying Boyle's Privacy Framework

Privacy is a generic term that means different things to different people. To enable the unambiguous description and discussion of privacy-related issues in media spaces, Boyle created a framework that collects various terms and concepts related to privacy from the literature. His framework (Boyle, 2005) broadly divides privacy into the three “control modalities” of *confidentiality*, *autonomy*, and *solitude*. Confidentiality relates to a person’s control over what information others can access about him/her; autonomy relates to how a person chooses to present him/herself; and solitude relates to how a person controls interpersonal interactions.

In his framework, Boyle additionally considers the *mechanics of privacy* (low-level details and properties of privacy) and particular issues relating specifically to *computers and privacy* (e.g., security or user interface concerns). Boyle then defines a process where people can use his framework to systematically analyze particular systems (or situations or techniques, etc.). His method of privacy analysis consists of the following four steps (Boyle, 2005): 1) *partition*, where the analyst determines the primary focus of the analysis and distinguishes between main and secondary effects; 2) *describe*, where the analyst uses Boyle’s framework and vocabulary terms to unambiguously describe each aspect of the system; 3) *reveal*, where the analyst identifies assumptions that have been made with respect to the system; and 4) *summarize*, where the analyst summarizes the findings from the previous two steps, including an overview of “merits and demerits” of the system as well as the conditions in which these merits and demerits exist. In the remainder of this section, I analyze the Screen Sharing item using Boyle's privacy framework and method of privacy analysis.

6.1.1 Analysis Step 1: Partition

Boyle’s framework mainly considers privacy from the perspective of how it affects *confidentiality*, *autonomy*, and *solitude*. Within this context, the function of the Screen Sharing item is to share the contents of a person’s screen with other group members. Thus, the Screen Sharing item primarily affects confidentiality, the component of privacy that relates to a person’s control over what information others can access about him/her. For example, much information can potentially be revealed about a person from the contents of their shared screen, such as what they are working on, who they are instant messaging with, or what movie they will be seeing later that evening. Consequently, I primarily examine the Screen Sharing item's effect on confidentiality in this analysis, as this attribute presents the greatest risk to one's privacy.

However, it is important to recognize that the Screen Sharing item also secondarily affects autonomy and solitude. Autonomy is the component of privacy that relates to how a person chooses to present him/herself. The "observable manifestations of the self" (Boyle and Greenberg, 2005a) such as appearance, identity, and impression that a person uses to present him/herself to others can be affected by the information accessible about that person. For example, seeing someone's credit card or bank account balance on their shared screen may influence others' impressions of that person. Solitude is the component of privacy that relates to how a person controls interpersonal interactions. The Screen Sharing item can affect a person's solitude, for example by showing others that the person is not busy working on their computer, and thus potentially available for interruption. In this chapter, I also consider the Screen Sharing item's effects on autonomy and solitude, although not to the same depth as confidentiality.

6.1.2 Analysis Step 2: Describe

In this section, I systematically describe the main and secondary effects the Screen Sharing item has on its users' privacy.

Main effect: Confidentiality

As a tool for providing artifact awareness through screen sharing, the Screen Sharing item strongly affects confidentiality. It shares what a person is doing on their computer through the *visual information channel*, obtaining this information by *sampling*, i.e., by periodically taking a screenshot of the person's screen to distribute. From the shared screen image, much information can potentially be seen or inferred; some examples are shown in Table 6.1. The information being shared can be *sensitive* (e.g., personal communications or financial records) or not (e.g., publicly available information). While the expectation is that work-related artifacts or activities will have low sensitivity with respect to other members of the same group, it is possible that a person's work may be highly sensitive if that person is working on a ‘secret’ project, if viewers are not normally a part of the group, or if a viewer is in a position of authority.

The Screen Sharing item offers a number of controls for mitigating confidentiality concerns; these controls alter the *fidelity* of the information being shared so that some details are obscured. Using the privacy controls, people can change the *capture* region of the screen to include or exclude certain windows or areas. They can change the zoom level / resolution of the shared screen image to make text legible or illegible. They can change the frequency at which the shared screen image is being updated to change the timeliness of the information being shared. They can also distort the shared screen image through blurring or pixelization in an *edit* operation on the image before it is transmitted to others. These privacy controls were described in full detail in Section 3.4. However, these brief descriptions illustrate that the privacy controls in the Screen Sharing item focus specifically on changing the *precision* of the information being shared, rather than the *accuracy*; the shared screen image always shows (with differing amounts of detail) exactly what is on a person’s screen in the region being captured.

Still, even though the accuracy of the shared information does not change, *disinformation* is not hard and *misinformation* is possible; for example, the shared screen image could intentionally or unintentionally show that a person is working on writing a paper when in reality they are surfing the web in an area of the screen not being shared.

Information Topic	**Examples**
Information about the self	a calendar or schedule; what a person is working on or looking at
Personally identifying information	an IM window with the person’s name; a unique desktop background; the Screen Sharing item also identifies which person’s screen it is sharing
Activities	open application windows can indicate that a person is coding, surfing the web, writing a report, communicating through IM, etc.
Whereabouts	windows moving around, etc. indicate a person is at their computer; calendar/schedule information shows where the person is or will be; the Screen Sharing item darkens if a computer becomes idle for 5 minutes
Encounters	text in a group IM; videos of participants in a conferencing call
Utterances	text in IM or email messages
Actions	over time, typing, scrolling of windows, opening or closing of applications
Relationships	names in an IM contact list or email inbox

Table 6.1 – Information about a person that can be revealed through the Screen Sharing item.

Thus, there is still some *plausible deniability* about what a person is doing on their computer. However, it is possible for the Screen Sharing item to reduce the plausible deniability of IM or email. For example, if others can see a person working on their screen, that person may no longer be able to pretend that they are ‘away’ from their computer, or if others can see a person read a message or an email, that person can no longer pretend that they haven’t received it or seen it yet.

The information shared by the Screen Sharing item typically has low *persistence*; on auto-update, a new shared screen image is captured and replaces the previous image at least once every 90 seconds. When a user logs off of Community Bar (CB), their shared screen image is automatically removed from CB – the images are not stored anywhere for *archival* purposes. However, it is possible for someone to make a persistent copy of a shared screen image by taking a screenshot of it and saving it to file. It is also possible for someone to hack into the underlying distributed data structure in order to automate screen recording. There is no support in the Screen Sharing item for tracking any *use*, *misuse*, or *misappropriation* of the information being shared. There is also nothing to prevent the *scrutiny* or *surreptitious surveillance* of what one is doing on one’s computer, particularly by one’s superiors (a common concern). Still, the Screen Sharing item does provide some feedback about what others can see and whether or not others are currently looking closely at one’s screen, as well as ways to reduce the amount of information others can gain from a shared screen image (at the cost of also reducing awareness).

Secondary effects: Autonomy and solitude

The Screen Sharing item also has secondary effects on autonomy and solitude through their relationships to confidentiality.

Autonomy is related to confidentiality in that people can present themselves in a particular way by controlling what information about themselves others can access. Thus, the information that is shared through the Screen Sharing item about what a person is doing on their computer can influence others’ *impressions* of that person. In particular, if a person works from a remote location where one of the only information channels about them is their shared screen, they may be judged by others primarily based on what their shared screen shows. This can become a problem when *back-stage performances* mistakenly occur on the *front-stage*, such as when someone working from home forgets that they are sharing their screen and starts looking at videos that are inappropriate for a work environment, though acceptable for watching at home. Privacy *harms* such as these that relate to autonomy can be *aesthetic* (e.g., a person often seen watching videos online instead of working being thought of as unproductive) or *strategic* (e.g., that same person being passed over for a promotion because they are perceived as being unproductive).

The Screen Sharing item aims to prevent mistaken back-stage performances from occurring on the front-stage by providing appropriate warning or reminder before a screen capture is taken. Also, since the expectation is that the Screen Sharing item will be used by work peers, it is assumed that *trust* has already been established between the group members who will be sharing their screens with one another and that group members will work to maintain that trust as well as help others preserve their privacy if needed, e.g., audience members can use the Chat item in CB to warn the person sharing their screen about things that the person may not want to be sharing.

Like autonomy, solitude is also related to confidentiality; people can somewhat control their interactions with others by controlling what information about themselves others can see. For example, the Screen Sharing item shows what a person is doing on their computer. This information can be used to determine that person's *availability* and also sometimes that person's *accessibility*, which can help others determine whether that person can be interrupted.

Part of how people control their interactions also involves how they focus their attention and how they react to distraction. The Screen Sharing item typically shares information about others' screens though its tile in the bar, which sits on the *periphery* of the user's *attention*. If more information is desired, the user can see additional detail in the tooltip grande or full view, or can transition into *focused* interaction by engaging in a remote pointing session with another user. The information others are sharing may not always be of direct *relevance* to one's work. However, the idea is that valuable interactions can still be triggered by 'unimportant' artifacts on someone's screen.

Mechanics of privacy

The Screen Sharing item affects many of the boundaries across which privacy is regulated. It allows regulation of the *disclosure boundary* by offering controls that people can use to reveal or obscure information on their screen. It enables people to transcend *spatial boundaries* by helping them share their individual work with others across distance. Because of the distance however, regulation of the *identity boundary* is more complicated, since the Screen Sharing item makes it possible for problems like back-stage performances on the front stage to occur.

Because the Screen Sharing item shows what a person is currently doing on their computer, the *possibility* for various kinds of privacy *violations* of varying *probability* and *severity* exists. One example of a low probability, high severity privacy violation is a *deliberate abuse*, where a staff member from a rival company manages to intercept the shared screen data being sent to or from the shared dictionary in order to gain information for competitive advantage. More examples of possible privacy *problems* are described in Section 3.1.2 and in Table 6.2.

To manage these potential privacy problems, people can use a number of strategies. Through *self-appropriation* or self-scrutiny, people monitor what they are sharing and manage the impression that they give to others. Part of self-appropriation can be the use of *disinformation*, for example to make it seem to others that they are working when they are not. However, self-appropriation can be cognitively draining as it takes effort to continually monitor what is being shared and to decide what is appropriate for being shared (Boyle and Greenberg, 2005a). The *risk/reward trade-off*, i.e., the balance between privacy and awareness, can be managed through both technological and social controls. Technological controls include the privacy controls that were explicitly built into the Screen Sharing item, which people can use to reduce the amount of information they are sharing at the cost of also reducing awareness. Social controls include *self-policing* by the group.

The Screen Sharing item also provides various types of *environmental support* for privacy; for example, the Screen Sharing item supports *reflexive interpretability of action* by providing similar tile and tooltip grande views to the owner as to the audience, as well as by showing previews in the owner's full view of what others will see if the owner changes various privacy settings. *Constraints* on what information is being shared can be placed on the Screen Sharing item – people can make a *choice* about how much information they want to show others by configuring the various privacy control settings or even by deciding not to share their screen at all. *Cues* on what information is being shared are provided through *feedback* of what one is sharing and when others are looking at one's screen. *Refuge* can be somewhat gained by putting the Screen Sharing item on manual updates. However, there is no *feedthrough* or natural *transitions* between different levels of privacy, and *reciprocity* is not enforced, which may lead to *risk/reward disparity* since a person revealing more of their screen or more details of their screen (thus increasing the risk for a privacy violation) may not receive proportional benefit / reward since others may not have to share the same level of detail. *Situated action* is also not supported directly by the Screen Sharing item, although over time, people can build up information about the context of someone's work by seeing their screens or through interaction with them.

Computers and privacy

The Screen Sharing item does not have any *access control* or *authorisation*; anyone who joins the group in CB can see all the screens being shared. The only *content control* the Screen Sharing item provides is the *distortion filtration* of the shared screen images. A way of blocking out certain regions of the screen so that they can't be seen by others or would be seen in different ways by different people could be added; this technique has been implemented and extended by Berry et al. (2005) and is a form of *publication filtration*.

Some examples of privacy harms were given in Section 3.1.2. In Table 6.2, I articulate other potential privacy problems from Boyle's framework that could occur from use of the Screen Sharing item. While some of these problems are considered in the current version, others are left for future work to address.

Privacy Problem	**Common Cause**	**Example**
Inadvertent privacy infraction	People forget that their screen is being shared.	Lauren forgets that she is sharing her screen when she starts working with confidential data and inadvertently shares it.
Apprehension	People do not want to make a bad impression in front of others.	Mitch does not want to share his screen because he is afraid others might see him reading comics online during the day and think he doesn't work very hard.
Resentment	People resent a loss of control over their own privacy.	Lauren feels resentful that she is required to share her screen so that her boss can monitor her work, an implied lack of trust.
Decontextualisation	People share their screen or a region of their screen without its surrounding context.	Mitch is sharing his web browser, which currently contains comics. Others may think that he is taking a break, but what they don't see is that in another region of his screen, Mitch is creating a presentation that includes slides on visual cues and he is looking at comics for examples to include.
Disembodiment	People share their screens from different locations / environments than their distant colleagues.	Lauren, sharing her screen from home, forgets that others on CB are working from the office and she starts looking at some photos taken at a wild party last weekend, photos inappropriate for a work setting.
Dissociation	The Screen Sharing item has no explicit way to identify which audience members are looking at a shared screen.	Mitch notices that someone is looking at his shared screen in detail in their full view, but he does not know which of the ten people currently on CB that person is.
Role conflict	People play different roles in different social worlds and sometimes previously non-overlapping social worlds "collide".	Lauren receives an email from her friend with photos from the weekend party and starts looking at them at work, though the photos turn out to be inappropriate for a work setting.
Misappropriation	People can be competitive and might take any opportunity to advance their career.	Mitch sees on Lauren's shared screen that she has come up with a solution to a problem the team has been working on; he quickly puts together an email to their boss in which he takes credit for the solution.
Misuse	People might take any opportunity to make personal or financial gain.	Lauren sees non-public information on Mitch's shared screen and based on that, sells some of her company stock.
Identity theft	People might take any opportunity to make personal or financial gain.	Mitch sees shopping information on Lauren's shared screen and copies down her credit card number and expiry date for future use.
Impersonation	People might act maliciously towards others they dislike or might take any opportunity to make personal or financial gain.	Lauren has some of Mitch's personal information from previously seeing it on his shared screen, and cancels his internet service at his home in order to disrupt his life.

Table 6.2 – Potential privacy problems that might occur through use of the Screen Sharing item.

6.1.3 Step 3: Reveal

Several assumptions have been made in the design of the Screen Sharing item, as well as about how it and CB are used by a community. First, there is an assumption that the Screen Sharing item will be used by groups of people who trust and collaborate with each other. Thus, no explicit precautions to prevent privacy problems such as deliberate abuses have been taken. The Screen Sharing item would not work in a competitive environment, except perhaps as a source of disinformation; in fact, it has not been designed for a competitive environment at all.

Secondly, there is an assumption that people using the Screen Sharing item are doing so voluntarily, in the sense that they will be motivated to share parts of their screen useful for others to see. If people are forced to use the Screen Sharing item when they don't want to (e.g., through peer pressure or because upper management wants to monitor them), they can share parts of their screen that are not useful to others, e.g., an empty area of their computer desktop, or they can use the privacy controls in the Screen Sharing item to obscure information or disinform others. In other words, people can easily get around a requirement to share their screen, at the expense of benefits for awareness.

Another assumption is that inadvertent privacy violations are the ones that people are most readily apprehensive about; this is why the privacy controls currently built into the Screen Sharing item seek primarily to address these kinds of violations. Related to this is an assumption that the high precision details of a shared screen image, for example the text in a document or an email, are what make the contents of a shared screen sensitive. If these details are distorted, such as by blur filtration, another assumption is that privacy will then likely be preserved. However, the high-precision details may not always be what makes a shared screen image sensitive; for example, if a person were playing a game or looking at inappropriate videos or pictures at work, this may still be obvious from the overall colours and shapes seen in a shared screen image, even if the image were distorted. In fact, blur filtration has been shown to fail to preserve privacy in some high-risk situations in a home media space (Neustaedter et al., 2006), such as when people are caught by the camera in a state of undress. Some of the snapshots in these high-risk situations can be similar to pictures or videos that people might look at on their computer that are then unintentionally captured and broadcast by the Screen Sharing item. In other words, blur filtration can work well in cases where the text is the sensitive part of a shared screen, but can fail to preserve privacy when an image or a visual symbol is the part of the shared screen content that is sensitive.

6.1.4 Step 4: Summarize

In this section, I summarize the findings from this theoretical analysis of privacy. As seen, the Screen Sharing item is a prototype artifact awareness tool used between intimate collaborators to share their individual work and work artifacts. Because people are sharing their screens with each other, there exists particular risk that sensitive information will be revealed. To counter this risk, the Screen Sharing item includes several privacy controls, including blur filtration, for changing the precision of the information being shared. These controls can help prevent inadvertent privacy infractions from occurring when it is the high-precision details that make the shared information sensitive (while still providing useful awareness information through the low-precision details). However, when the low-precision details contain the sensitive information, the controls in the Screen Sharing item no longer help preserve privacy.

There are also other privacy problems that the Screen Sharing item does not currently handle well, such as deliberate abuses. However, for expected use by a small group of intimate collaborators who work together such as the groups discussed in the previous chapter, the existing controls seem to be ‘good enough’ most of the time for preserving users’ privacy and alleviating apprehension users may have about using the system.

6.2 Discussion

In this section, I first revisit some of the findings from the informal evaluations (discussed in the previous chapter) and relate them to findings from the theoretical analysis done using Boyle’s method of privacy analysis. I then use these findings to hypothesize about how other groups might use the Screen Sharing item. Finally, I conclude by briefly discussing opportunities for future work relating to privacy in the Screen Sharing item.

6.2.1 Revisiting Privacy as Seen in the Informal Evaluations

As seen in the informal evaluations, privacy was less of a concern than expected for the participants in both groups that used the Screen Sharing item. This may partly have been because of the following reasons: both groups consisted of intimate collaborators who trusted and sometimes worked directly with one another; participation in the informal evaluation was voluntary so people who were apprehensive about sharing their screens simply did not share them; few participants shared their screens from their home computers when they were not doing work; and as new users of the Screen Sharing item, participants were also likely on their 'best' behaviour and were careful with what they were sharing.

Of the privacy problems in Table 6.2, only instances of inadvertent privacy infractions, apprehension, dissociation, and role conflict were reported. These instances are recapped in Table 6.3, and were the most serious privacy problems encountered by participants (that they were willing to report). Deliberate privacy abuses seemed to be non-existent for both groups.

Privacy Problem	**Example**	**Cause**
Inadvertent privacy infraction	A participant who was creating exam questions almost accidentally shared them.	The participant forgot that their screen was being shared.
Apprehension	A CB group member did not want to share their screen, though they remained part of the CB community during the study.	The group member did not want to make a bad impression in front of others.
Dissociation	Several participants commented that they would have liked to know the identities of the people looking at their screen in the full view, not just that someone was looking.	The Screen Sharing item currently does not identify which audience members are looking at a shared screen.
Role conflict	A participant working at an industrial site on product development initially shared confidential code at full clarity.	The participant was acting both as a company employee and as a CB member, though the other CB members weren't employees of the same company.

Table 6.3 – Privacy problems seen in the informal evaluations with two groups.

To protect privacy, the most commonly-used strategy was to separate semi-public information and private information onto different displays. This strategy took advantage of the physical environment and hardware setup, and could also be used for disinformation to make it seem to others that the owner was working when the owner was actually doing something else. Another strategy people used was to blur their shared screen images to change the precision of the information being shared. During the informal evaluation, this distortion filtration seemed to work well as the information considered sensitive was mostly text (e.g., communications, exam questions, code). None of the participants reported situations in which blur filtration failed to preserve their privacy; however, it is important to note that participants were only concerned with keeping text illegible – they did not mind others knowing what they were doing.

In summary, the findings from the informal evaluation suggest that when the assumptions discussed in Section 6.1.3 hold, the Screen Sharing item can successfully be used with few privacy problems. In the next section, I discuss two examples of how use of the Screen Sharing item might differ for other groups.

6.2.2 Generalizing Privacy to Other Groups

From the theoretical analysis, it can be seen how privacy risks from use of the Screen Sharing item might change if the assumptions outlined in Section 6.1.3 fail to hold. For example, if the Screen Sharing item were used by employees who were not intimate collaborators but competitors, there would likely be more deliberate abuses of privacy seen. In this section, I briefly speculate on how groups with differing technology or differing social dynamics might use the Screen Sharing item differently from what was seen in the informal evaluations.

A group of intimate collaborators with a single-monitor setup

Small differences in technology, seemingly unrelated to privacy, could potentially affect privacy in large ways. As an example, consider a group of intimate collaborators similar to the groups used in the informal evaluations, except with a *single*-monitor setup. I choose this group to discuss in more detail as it seems plausible that such a group might adopt use of the Screen Sharing item. Also, Boyle's framework does not explicitly examine the effects of the hardware setup (independent from the system) on privacy.

In the informal evaluations, most participants had a dual-monitor setup. The dual-monitor setup seemed to be important in alleviating people's apprehensions about sharing their screens, which in turn contributed to the success as seen of the Screen Sharing item. However, many people still only have a single display (Hutchings et al., 2004). With only one display, inadvertent privacy violations would likely increase, as there would be less display space available and the Screen Sharing item shares information from whatever window is 'on top' in the capture region. For example, if an instant message appeared, part of it could easily be placed in the capture area of the screen and be unintentionally shared.

Another consequence of limited display space is that disinformation could become more difficult. While it would still be possible to pretend to be doing work while actually doing something else, the setup for the disinformation would likely be more complex (e.g., windows would have to be manually resized and placed so that they do not overlap) and the activities outside of the disinformation would likely be more difficult to do (e.g., allocating enough display space to convincingly disinform others means that there would be less available display space for other activities).

Still, although some strategies for preserving privacy while using the Screen Sharing item may become less effective for users with a single-monitor setup, other strategies might remain useful. For example, distortion filtration could be used to decrease the risk of inadvertently sharing sensitive text information, and setting the Screen Sharing item to only share manual updates could be used to help disinform others about what is on one's screen. Other features that could be added to the Screen Sharing item to help support single-monitor users in particular will be discussed in Section 6.2.3.

A group of staff members required to share their screens by their boss for monitoring

In contrast to the previous example, consider a group with a different social structure, such as a group of staff members who are required to share their screens at full fidelity with their boss so that their boss can monitor their work. While the Screen Sharing item is intended as an awareness tool for use by intimate collaborators and peers, it is possible that it may be abused by someone in a position of power.

Faced with this implied lack of trust from their boss, staff members might feel resentful about having to use the Screen Sharing item. This resentment may lead to them sharing only 'useless' parts of their screen, or even to them sabotaging the system. Disinformation would likely be highly used as a way for staff members to appear to be working while actually doing something else. Even when actually working, they may not feel inclined to share their work with a boss who feels the need to monitor them.

Staff members' work time would be spent undermining the system in an effort to disinform their boss rather than actually doing work. Their boss would find the Screen Sharing item an ineffective tool for monitoring subordinates. In a situation like this, the Screen Sharing item could be harmful to the work environment and group dynamic while providing little, if any, benefit.

6.2.3 Opportunities for Future Work Relating to Privacy

The omissions revealed by applying Boyle's framework to the Screen Sharing item suggest several possible additions for supporting users' privacy. While it is beyond the scope of this thesis to detail all possible changes, I discuss a few examples here.

First, the Screen Sharing item currently only allows people to change the fidelity of the information being shared, not the accuracy. While distorting the accuracy of shared information can easily lead to misinformation or disinformation, it can be useful for times when distorting the fidelity is not enough to preserve privacy. Possible ways for changing accuracy might use image-processing algorithms to replace certain colours, shapes, or text in a shared screen image with other (possibly random) colours, shapes, or text.

Additional ways for altering the fidelity of shared information would also be useful; one of the participants in the informal evaluation requested a feature for blocking out certain regions of his screen so that they couldn't be seen by others. This technique, which has been implemented and extended by Berry et al. (2005) into a role-specific view-masking technique, might be particularly useful for people who only have one screen. The idea is that the owner can selectively mask portions of his/her screen image on a per person basis, where others may see different things depending on the settings.

The Screen Sharing item also lacks some of the environmental support that Boyle's framework includes. Adding natural transitions between different levels of privacy could make balancing privacy and awareness easier for users; for example, the Screen Sharing item could detect when a person is sharing an email or IM window (e.g., by recognizing the program name) and automatically blur the shared screen image.

Finally, while the Screen Sharing item is currently able to share a user-specified window, if other windows overlap this window, the information from these windows will be captured, thus leading to increased risk of inadvertent privacy infractions. Modifying the implementation so that regions of the specified window that are covered will be blocked out will help decrease the risk of inadvertent privacy infractions, particularly for single-monitor users who have more limited display space in which to fit all their windows. This technique has already been implemented by application sharing systems such as MSN Messenger's application sharing tool.

6.3 Summary

As a tool that shares the contents of a person's screen with other group members for awareness, the Screen Sharing item primarily affects confidentiality, the component of Boyle's privacy framework that relates to a person's control over what information others can access about him/her. From a person's use of the Screen Sharing item, much information about that person can be potentially be seen or inferred, such as information about the person's activities, whereabouts, or relationships with others. To help alleviate apprehension and prevent privacy problems, the Screen Sharing item offers several controls for changing the fidelity of the information being shared. These controls can also be used to support disinformation.

As part of applying Boyle's privacy framework to the Screen Sharing item, I identified several assumptions made in its design. First, there is the assumption that the

Screen Sharing item will be used, voluntarily, by groups of intimate collaborators. Second, there is an assumption that people will be primarily concerned about inadvertent privacy violations. Finally, there is the assumption that the high-precision details of a shared screen are what make the contents of the shared screen sensitive. When these assumptions hold, as in the informal evaluations, the Screen Sharing item can successfully be used with few privacy problems. If these assumptions fail, then new privacy risks and problems may be introduced.

In this chapter, I also briefly discussed potential areas of future work relating to privacy. In the next chapter, I conclude this thesis by summarizing its contributions and by outlining more general areas of potential future work.

Chapter 7. Conclusion

In this thesis, I investigated the use of screen sharing for supporting artifact awareness leading to casual interaction between intimate collaborators. In particular, I described the design and development of a screen-sharing awareness tool (Chapters 3 and 4), reported on its evaluation with two groups of intimate collaborators (Chapter 5), and examined how it can affect its users' privacy in a theoretical analysis (Chapter 6). In this final chapter, I revisit my original research questions, summarize my thesis contributions, and conclude by discussing possible directions for future work relating to this research.

7.1 Research Questions Revisited

In Chapter 1, I raised the following three research questions relating to the area of computer-supported cooperative work:

1. *How can awareness of ongoing individual work be increased between intimate collaborators, particularly members of a distributed group?* Existing informal awareness and casual interaction tools generally only provide awareness of people; few systems share the artifacts that people are working on without them having to move to a "meeting" mode.

2. *What mechanisms can help people control how much information they reveal to others? What mechanisms can let others control how much information they receive?* For someone sharing their individual work with others, the challenge for an awareness tool is to provide information about that person's work and activities while still supporting that person's privacy needs. Similarly, an awareness system must also consider that people receiving this information may want access to

varying amounts of awareness information in different situations. Yet, it is uncertain how enabling these should be done.

3. *How do such systems work in practice?* While there have been prior studies of how existing informal awareness and casual interaction tools are used by various groups, existing tools generally do not provide awareness of ongoing individual work. Consequently, we do not know how such an awareness system would be adopted and used in practice.

7.2 Thesis Contributions

These research questions were addressed through the following three main research contributions:

1. *Screen-sharing awareness tool.* In Chapters 3 and 4, I presented the design and implementation of an awareness tool that uses screen sharing within the Community Bar to increase awareness of ongoing individual work between groups of intimate collaborators (addressing Question 1). While awareness systems that use screen sharing do exist (e.g., Gutwin et al., 2005; Rounding, 2004), they were intended as proof-of-concept prototypes and do not consider in detail how to provide awareness using screen sharing. My screen-sharing awareness tool has been designed to address issues such as privacy and distraction, and has also been developed to meet a level of performance sufficient for its expected use so that it can be deployed and evaluated by end-users.

2. *Privacy controls and feedback; different levels of information and interaction.* In Chapters 3 and 4, I also described the design and implementation of privacy controls and feedback mechanisms for people sharing their screens as well as different levels of information and interaction for viewers (addressing Question 2). People sharing their screens can specify what they want to share with others, how often they want to share with others, and how much detail they want to allow others to see. Feedback mechanisms remind them that their screen is being shared as well as indicate when others are looking at their shared screen in detail. Viewers can see

thumbnails of shared screens in their bar or in the expanded view, they can zoom and pan in the larger view to see shared screens in more detail, and they can discuss and point at artifacts in the remote pointing view. To partially evaluate the success and failure of this design in addressing Question 2, Chapter 6 reports on a theoretical analysis of privacy that I did by applying Boyle's privacy framework to the screen-sharing awareness tool. In this analysis, I articulated possible privacy problems for potential users, which include apprehension and inadvertent privacy violations. I showed that when the screen-sharing awareness tool is used voluntarily by groups of intimate collaborators in situations where the sensitive details of a shared screen's contents are in the high-precision details, the screen-sharing awareness tool can be used successfully with few privacy problems. I also showed that when these conditions are not met, new privacy risks and problems may be introduced.

3. *Evaluation with end-users*. In Chapter 5, I discussed findings from an evaluation of the screen-sharing awareness tool with two different groups (addressing Question 3). One group was an internal research lab that had already been using Community Bar on a daily basis for over a year, and the other group was an external commercial development team, which had been introduced to Community Bar and the screen-sharing awareness tool at the same time. Based on discussions with these two groups, I found that the majority of uses were to maintain awareness of what others were doing and to opportunistically share computer artifacts or activities with others. I also observed that: the screen-sharing awareness tool seemed particularly useful for certain subsets of the group, such as people who were working closely together and people who were good friends; the screen-sharing awareness tool helped people engage in lightweight causal interaction by allowing people to check availability or share artifacts onscreen without having to leave their desks; and, privacy was less of a concern than expected, which I further discussed in Chapter 6.

7.3 Future Work

The work in this thesis is the beginning, rather than the end, of a long-term research program. While my work suggests what could be, there is considerable room for improvement in the system design, in our understanding of what people really want in terms of artifact sharing and awareness, and in evaluating the use and cultural adoption of such systems. Consequently, possible future directions this research could take include improving on the current approach of using screen sharing in the Community Bar to provide artifact awareness to groups of intimate collaborators; further evaluation of how the screen-sharing awareness tool can be adopted for use by different groups; and, exploration of alternate ways of using screen sharing for providing artifact awareness.

7.3.1 Improvements to the Current Design

There are several obvious ways that the current design of the Screen Sharing item in Community Bar could be improved.

First, the system functionality needs to address all aspects of the communications life cycle. For example, the Screen Sharing item could be linked to a full screen sharing system, in order to provide a more complete transition from awareness to full groupware. This was done in the Notification Collage Desktop media item (Rounding, 2004). Also, the only communication channel currently built into the Screen Sharing item is text chat. When people are involved in a remote pointing session in which they are using text chat to discuss shared artifacts, they must switch back and forth between using the keyboard (to communicate) and the mouse (to point at things). To let people interact more naturally when discussing shared artifacts, an audio channel could also be added to the Screen Sharing item.

Second, the system needs to provide better information. One of the most common requests from participants was a way to identify which audience member(s) were looking closely at one’s shared screen. In the real world, we see others approach, lean into our workspace, and glance at our artifacts. It is easy to tell who they are, how closely they are looking, where they are looking, and so on. This is not supported well in the current system, in which the Screen Sharing item only indicates that *someone* is looking closely at a shared screen and not *who* is looking. One possible approach is to supply additional information, e.g., an equivalent of the glance feature in Montage that shows that people are about to look in and that identifies them (Tang and Rua, 1994). It would be interesting to see whether identifying people looking at a shared screen in detail changes how willing people are to look at shared screens in the larger views. While the owner currently has some idea of who the audience member looking at their screen may be (since the Screen Sharing item is only visible to others in the same CB place as the owner), audience members currently have plausible deniability that they are looking at a particular shared screen. Also, the accuracy of the feedback the owner receives could be improved, so that the feedback is only given when the shared screen is visible to the audience member, i.e., if the large view is open but is covered up by another window or is minimized, remove the feedback to decrease the instances in which the owner thinks that an audience member is scrutinizing his or her shared screen when the audience member is actually not doing so.

Finally, Section 4.3.4 describes ways that the performance of the screen-sharing awareness tool could be improved, and Section 6.2.3 describes ways that preserving users' privacy could be improved. Implementing these recommendations would help make the system less awkward in terms of its response times and its interface controls.

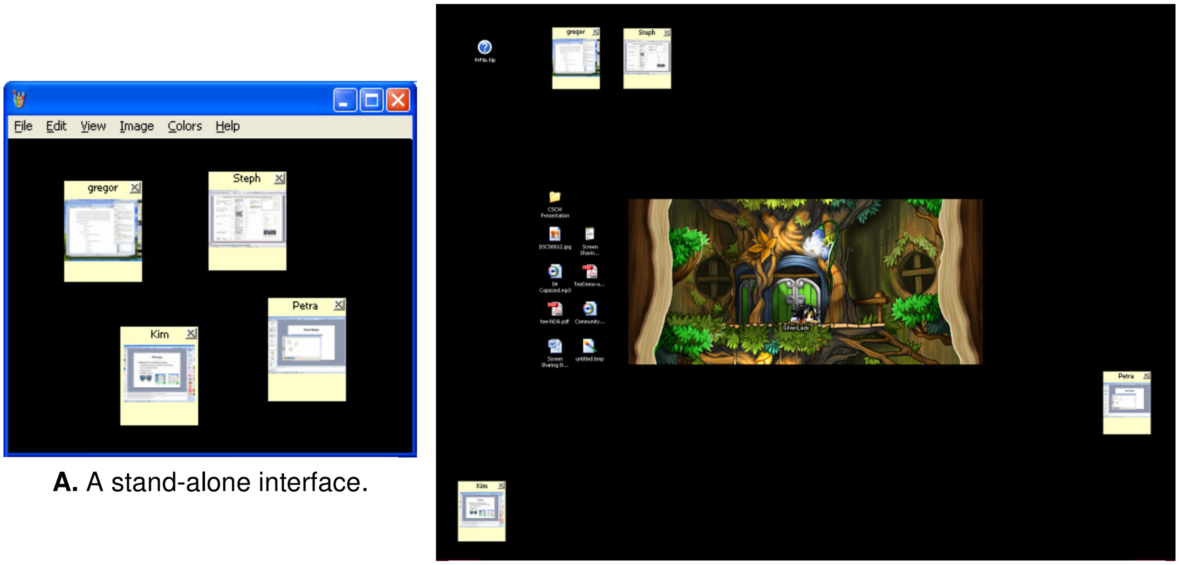
7.3.2 Further Evaluation

In Chapter 5, I described how the Screen Sharing item was informally evaluated by two groups of users. Future work should include deploying the system to additional groups for longer periods of time to formally observe how they adopt the screen-sharing awareness tool for their own use. These additional groups could be ones similar to those hypothesized about in Section 6.2.2, to test my hypotheses of expected use in those types of groups. Interesting questions include: How does use of the screen-sharing awareness tool change over time or between different groups? What are the effects of reciprocity on using shared screens for awareness – are people still willing to share their screens with others if those others never or rarely share their screens? Does the screen-sharing awareness tool actually improve productivity or improve people's working conditions? An analysis of the screen-sharing awareness tool with respect to how well it meets Grudin's eight challenges for groupware developers (Grudin, 1994) would also be informative, and such an analysis may reveal additional success / fail conditions for the system as well as suggest additional improvements for supporting groups.

Longitudinal studies are also critical, as this technology falls under what are called 'socio-techical systems' – their acceptance and use is as much about the culture that develops around it as it is about the features that the software provides. Screen sharing of one's desktop is still a strange concept, even though the virtual desktop superficially resembles one's physical desktop and how it is seen by others in an open office. The culture of use that develops around this technology could lead to outright rejection, outright acceptance, or (most likely) something between the two. The expectation is that people will find situations where tools such as these are valuable, and adapt their work behaviours around them. It is this adaption that is extremely interesting but difficult to probe in short-term studies.

7.3.3 Exploration of Alternate Approaches

Finally, sharing screens within the Community Bar is only one way of providing artifact awareness through screen sharing. Future work could include exploring different ways of presenting such screens to people. For example, a stand-alone interface could be used where tiles of the shared screens are scattered around in a window (Figure 7.1a), much like the interface to the Notification Collage. Or, perhaps shared screens could be presented in gadgets that could be placed and moved around on a desktop (Figure 7.1b). The mechanism of initiating shared screen awareness between people could also be quite different. The Community Bar has a strong notion of a 'group', but imagine instead a situation where one person simply sends a desktop icon to another person by dragging and dropping it into an IM chat window. The recipient accepts it, and it may appear on



A. A stand-alone interface.

B. Gadgets on a desktop.

Figure 7.1 – Possible alternative ways to present shared screens for awareness.

their desk as in Figure 7.1b as a live desktop icon that can be explored in depth. Perhaps the sender can 'recall' or 'disengage' that desktop sharing at any time through controls on their local view. This model is much more in keeping with how IM works, and may be more appropriate for ad-hoc groups. Or, it could be part of the IM client itself. Just as people can update their display name space in real time or change their photo (avatars), they may be able to post their desktop region to their community. The live icon would then appear next to their name in other people's contact lists.

Another interesting question to investigate is how changing the representations of artifacts in a shared screen might affect how easy or difficult it is for people to maintain artifact awareness. For example, rather than sharing an image of the artifact in a screen, consider sharing text describing what the artifact is (similar to how the current design shares a text description if the tile in the bar shrinks too small for an image). Or, consider sharing just the icon of the application currently being used. Or perhaps abstract colours or shapes could be used to represent artifacts associated with particular tasks, such as communication or code development. Changing the representations of shared artifacts in ways like these may eliminate some privacy and space-usage concerns while still providing enough information for people to maintain awareness of what others are doing.

7.4 Final Words

In this thesis, I have described the design, implementation, and evaluation of an awareness tool that uses screen sharing within the Community Bar to support artifact awareness between intimate collaborators. With the awareness tool, people see others' screens in miniature at the edge of their display, can selectively raise a larger view of that screen to get more detail, and can engage in remote pointing if desired. People balance awareness with privacy by using several privacy-protection strategies built into the system.

Screen sharing was originally created to give collaborators the ability to do focused, joint work across distance; the initial experiences people had reveal that this was one of the ways in which the screen-sharing awareness tool was used. Yet, these experiences also reveal the importance of screen sharing for artifact awareness. People used the screen-sharing awareness tool to maintain awareness of what others were doing, to influence others' impressions of them, to monitor progress and coordinate joint tasks, to help determine when another person could be interrupted, and to engage in serendipitous conversation and collaboration.

Artifact awareness is an important component of informal awareness that has not been well-supported in existing informal awareness and casual interaction systems. I hope that the research described in this thesis lays a foundation that will motivate others interested in providing informal awareness and casual interaction to groups to also include support for artifact awareness. The research described in this thesis is a starting point from which they can base their designs and intellectual investigations.

References

Bartle, R. (2004). **Designing virtual worlds**: New Riders.

Bellotti, V. and Bly, S. (1996). **Walking away from the desktop computer: Distributed collaboration and mobility in a product design team**. In Proceedings of the ACM conference on Computer Supported Cooperative Work (CSCW 1996). ACM Press, 209-218.

Berry, L., Bartram, L., and Booth, K. (2005). **Role-based control of shared application views**. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2005). ACM Press, 25-32.

Bly, S. A., Harrison, S. R., and Irwin, S. (1993). **Media spaces: Bringing people together in a video, audio, and computing environment**. Communications of the ACM, 36(1), ACM Press, 28-46.

Boyle, M. (2005). **Privacy in media spaces**. PhD Thesis, University of Calgary, Calgary, Alberta, Canada.

Boyle, M., Edwards, C., and Greenberg, S. (2000). **The effects of filtered video on awareness and privacy**. In Proceedings of the ACM conference on Computer Supported Cooperative Work (CSCW 2000). ACM Press, 1-10.

Boyle, M. and Greenberg, S. (2005a). **The language of privacy: Learning from video media space analysis and design**. ACM Transactions on Computer-Human Interaction, 12(2), ACM Press, 328-370.

Boyle, M. and Greenberg, S. (2005b). **Rapidly prototyping multimedia groupware**. In Proceedings of the 11th Int'l Conference on Distributed Multimedia Systems (DMS 2005). Knowledge Systems Institute, IL, USA.

Buxton, W., Fitzmaurice, G., Balakrishnan, R., and Kurtenbach, G. (2000). **Large displays in automotive design**. Computer Graphics and Applications, 20(4), IEEE Computer Society, 68-75.

Cameron, A. F. and Webster, J. (2005). **Unintended consequences of emerging communication technologies: Instant messaging in the workplace**. Computers in Human Behavior, 21(1), Elsevier, 85-103.

Clark, H. (1996). **Using language**: Cambridge University Press.

Deriche, R. (1990). **Fast algorithms for low-level vision**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(1), IEEE Computer Society, 78-87.

Dourish, P. and Bly, S. (1992). **Portholes: Supporting awareness in a distributed work group**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1992). ACM Press, 541-547.

Engelbart, D. and English, W. (1968). **A research center for augmenting human intellect**. In AFIPS Proceedings of the 1968 Fall Joint Computer Conference. AFIPS Press, 395-410.

Fish, R. S., Kraut, R. E., Root, R. W., and Rice, R. E. (1992). **Evaluating video as a technology for informal communication**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1992). ACM Press, 37-48.

Frederick, R. (1994). **Experiences with real-time software video compression**. Sixth International Workshop on Packet Video. Portland, Oregon.

Garfinkel, D., Gust, P., Lemon, M., and Lowder, S. (1989). **The SharedX multi-user interface user's guide, version 2.0**. STL-TM-89-07. Palo Alto, California: Hewlett-Packard Laboratories.

Greenberg, S. (1990). **Sharing views and interactions with single-user applications**. In Proceedings of the ACM SIGOIS and IEEE CS TC-OA Conference on Office Information Systems. ACM Press, 227-337.

Greenberg, S. (1999). **Designing computers as public artifacts**. International Journal of Design Computing: Special Issue on Design Computing on the Net (DCNet 1999), University of Sydney, November 30 - December 33.

Greenberg, S. and Rounding, M. (2001). **The Notification Collage: Posting information to public and personal displays**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2001). ACM Press, 514-521.

Grinter, R. E. and Palen, L. (2002). **Instant messaging in teen life**. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2002). ACM Press, 21-30.

Grudin, J. (1994). **Groupware and social dynamics: Eight challenges for developers**. Communications of the ACM, 37(1), ACM Press, 92-105.

Grudin, J. (2001). **Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2001). ACM Press, 458-465.

Gutwin, C., Greenberg, S., Blum, R., and Dyck, J. (2005). **Supporting informal collaboration in shared-workspace groupware**. HCI-TR-2005-01. Sakatoon, Saskatchewan, Canada: The Interactions Lab, University of Saskatchewan.

Handel, M. and Herbsleb, J. D. (2002). **What is chat doing in the workplace?** In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2002). ACM Press, 1-10.

Heath, C. and Luff, P. (1992). **Collaboration and control: Crisis management and multimedia technology in London underground line control rooms**. Journal of Computer Supported Cooperative Work*,* 1(1), Springer, 24-48.

Hutchings, D. R., Smith, G., Meyers, B., Czerwinski, M., and Robertson, G. (2004). **Display space usage and window management operation comparisons between single monitor and multiple monitor users**. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI 2004). ACM Press, 32-39.

Isaacs, E., Walendowski, A., and Ranganthan, D. (2002). **Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2002). ACM Press, 179-186.

Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D. J., and Kamm, C. (2002). **The character, functions, and styles of instant messaging in the workplace**. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2002). ACM Press, 11-20.

Kaasten, S., Greenberg, S., and Edwards, C. (2002). **How people recognize previously seen web pages from titles, URLs and thumbnails**. In X. Faulkner, J. Finlay, and F. Detienne (Eds.), People and Computers XVI (Proceedings of HCI BCS Conference Series) (pp. 247-265): Springer Verlag.

Kraut, R., Egido, C., and Galegher, J. (1988). **Patterns of contact and communication in scientific research collaboration**. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 1988). ACM Press, 1-12.

Kurlander, D., Skelly, T., and Salesin, D. (1996). **Comic Chat**. In Proceedings of the International Conference on Computer Graphics and Interactive Techniques. ACM Press, 225-236.

Lauwers, J. C. and Lantz, K. A. (1990). **Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1990). ACM Press, 303-311.

Mantei, M. M., Baecker, R. M., Sellen, A. J., Buxton, W. A. S., Milligan, T., and Wellman, B. (1991). **Experiences in the use of a media space**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1991). ACM Press, 203-208.

McEwan, G. (2006). **Community Bar: Designing for informal awareness and casual interaction**. MSc Thesis, University of Calgary, Calgary, AB, Canada.

McEwan, G. and Greenberg, S. (2005). **Supporting social worlds with the Community Bar**. In Proceedings of the ACM SIGGROUP Conference on Supporting Group Work (GROUP 2005). ACM Press, 21-30.

McEwan, G. and Greenberg, S. (2006). **Community Bar Poster**: Banff, AB.

McEwan, G., Greenberg, S., Rounding, M., and Boyle, M. (2006). **Groupware plug-ins: A case study of extending collaboration functionality through media items**. In Proceedings of the International Conference on Collaboration Technologies (CollabTech 2006). IPSJ SIG Groupware and Network Services, 42-47.

Nardi, B. A. (1993). **A small matter of programming: Perspectives on end user computing**: MIT Press.

Nardi, B. A., Whittaker, S., and Bradner, E. (2000). **Interaction and outeraction: Instant messaging in action**. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2000). ACM Press, 79-88.

Neustaedter, C., Greenberg, S., and Boyle, M. (2006). **Blur filtration fails to preserve privacy for home-based video conferencing**. ACM Transactions on Computer-Human Interaction*,* 13(1), ACM Press, 1-36.

Nielsen, J. (1993). **Usability engineering**. San Diego, CA: Academic Press.

Richardson, T., Stafford-Fraser, Q., Wood, K. R., and Hopper, A. (1998). **Virtual Network Computing**. IEEE Internet Computing*,* 2(1), IEEE Educational Activities Department, 33-38.

Romero, N., McEwan, G., and Greenberg, S. (2007). **A field study of Community Bar: (Mis)-matches between theory and practice**. In Proceedings of the ACM SIGGROUP Conference on Supporting Group Work (GROUP 2007). ACM Press, 89-98.

Root, R. W. (1988). **Design of a multi-media vehicle for social browsing**. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 1988). ACM Press, 25-38.

Rounding, M. (2004). **Informal awareness and casual interaction with the Notification Collage**. MSc Thesis, University of Calgary, Calgary, Alberta, Canada.

Sarin, S. (1984). **Interactive on-line conferences**. PhD Thesis, MIT, Cambridge, Massachusetts, United States.

Smale, S. and Greenberg, S. (2005). **Broadcasting information via display names in instant messaging**. In Proceedings of the ACM SIGGROUP Conference on Supporting Group Work (GROUP 2005). ACM Press, 89-98.

Tang, J. C. and Rua, M. (1994). **Montage: Providing teleproximity for distributed groups**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1994). ACM Press, 37-43.

Tang, J. C., Yankelovich, N., Begole, J., Kleek, M. V., Li, F., and Bhalodia, J. (2001). **ConNexus to awarenex: Extending awareness to mobile users**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2001). ACM Press, 221-228.

Voida, A., Grinter, R. E., Ducheneaut, N., Edwards, W. K., and Newman, M. W. (2005). **Listening in: Practices surrounding iTunes music sharing**. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005). ACM Press, 191-200.

Wagstaff, S. (1998). **Animation on the web**: Peachpit Press.

Whittaker, S., Frohlich, D., and Daly-Jones, O. (1994). **Informal workplace communication: What is it like and how might we support it?** In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1994). ACM Press, 131-137.

WOS Data Systems. (1987). **Timbuktu: The next best thing to being there, user's guide (version 1)**.

Appendix A. Performance Testing Details

In this appendix, I report on additional details from the performance tests described in Chapter 4. First, I give timing data for the individual steps of capturing, blurring, and resizing a shared screen image. Then, I report data on the sizes of the shared screen images being distributed to the group. Finally, I show how I obtained the network speed measurements used in Chapter 4 to determine approximate update times for audience members.

A.1 Data on Capturing, Blurring, and Resizing Speeds

Tables A.1–A.3 show the average times taken for the individual steps of capturing, blurring, and resizing the shared screen image under the different conditions (the screens captured in the 1280 x 1024 condition are shown in Figure 4.5). Figure A.1, which summarizes this data in a bar chart, is shown below.

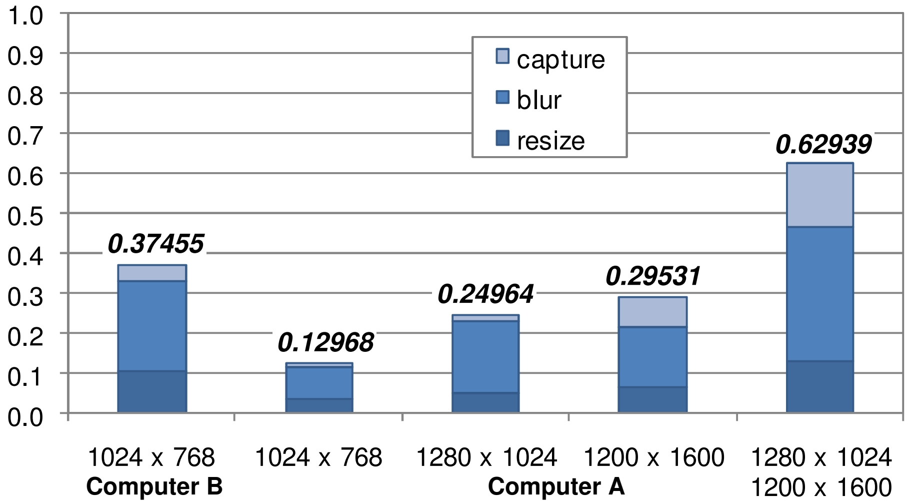


Figure A.1 – Total time in seconds to update the shared screen image locally, with individual times for capturing, blurring, and resizing the image shown (disregarding screen content).

Test Computer	Capture Area (pixels)	Screen Contents	Mean (seconds)	SD (seconds)
B	1024 x 768	code	0.04401	0.00170
		desktop	0.04330	0.00138
		web	0.04353	0.00168
A	1024 x 768	code	0.01105	0.00020
		desktop	0.01119	0.00040
		web	0.01100	0.00011
	1280 x 1024	code	0.01809	0.00052
		desktop	0.01806	0.00069
		web	0.01800	0.00023
	1200 x 1600	code	0.07673	0.00319
		desktop	0.07610	0.00169
		web	0.07646	0.00236
	1280 x 1024 1200 x 1600	code	0.15468	0.00263
		desktop	0.15957	0.01011
		web	0.16415	0.00979

Table A.1 – Average times taken to capture the shared screen image.

Test Computer	Capture Area (pixels)	Screen Contents	Mean (seconds)	SD (seconds)
B	1024 x 768	code	0.22973	0.00495
		desktop	0.22248	0.00502
		web	0.22327	0.00804
A	1024 x 768	code	0.08021	0.00364
		desktop	0.07902	0.00286
		web	0.08396	0.00678
	1280 x 1024	code	0.17501	0.00322
		desktop	0.17499	0.00275
		web	0.18162	0.00693
	1200 x 1600	code	0.14783	0.00533
		desktop	0.15192	0.00639
		web	0.14693	0.00492
	1280 x 1024 1200 x 1600	code	0.32311	0.00202
		desktop	0.33439	0.02285
		web	0.34563	0.02346

Table A.2 – Average times taken to blur the shared screen image.

Test Computer	Capture Area (pixels)	Screen Contents	Mean (seconds)	SD (seconds)
B	1024 x 768	code	0.11132	0.00975
		desktop	0.10290	0.00225
		web	0.10311	0.00240
A	1024 x 768	code	0.03785	0.00283
		desktop	0.03742	0.00181
		web	0.03733	0.00198
	1280 x 1024	code	0.05432	0.00277
		desktop	0.05423	0.00359
		web	0.05462	0.00181
	1200 x 1600	code	0.06993	0.00241
		desktop	0.06949	0.00255
		web	0.07053	0.00241
	1280 x 1024 1200 x 1600	code	0.13275	0.00213
		desktop	0.13540	0.00483
		web	0.13850	0.00733

Table A.3 – Average times taken to resize the shared screen image to a thumbnail.

As seen in Tables A.1–A.3, the times required for capturing, blurring, or resizing a screen image generally increase as the size of the original capture area increases. An exception to this is that blurring a 1200 x 1600 image appears to be faster than blurring a 1280 x 1024 image (Table A.2). It is not clear why this happens; perhaps the shape of the capture area in this condition (taller than it was wide, whereas the other areas captured were wider than they were tall) had an effect on how the algorithm processed the captured image, or perhaps there were more areas without text or images when the windows were resized to fill this larger space. Further tests would need to be done to determine the exact cause.

Also seen in Tables A.1–A.3, screen contents do not seem to have much effect on the times taken to capture, blur, or resize the shared screen images. In the next section, we will see that screen contents do have an effect on the compressed sizes of the shared screen images, but for Chapter 4, screen content is disregarded as a factor and averages over the different screen content conditions are used in the calculations for simplicity.

A.2 Data on Space Usage

Table A.4 shows the average measured sizes of the PNG-compressed thumbnail and full-size shared screen images distributed to audience members through the shared dictionary. An average over the screen contents for each different capture area was computed to simplify the calculations done in Section 4.3.3.

Test Computer	**Capture Area (pixels)**	**Screen Contents**	**Thumbnail (bytes)**		**Full Size (bytes)**	
			Mean	**SD**	**Mean**	**SD**
B	1024 x 768	code	18808	265	350017	4383
		desktop (some icons)	24023	26	527691	773
		web	25526	1261	528680	3415
	Average for 1024 x 768		***22689***	***2899***	***468796***	***84521***
A	1024 x 768	code	19237	315	337791	4550
		desktop (many icons)	32847	333	687001	5866
		web	27321	532	569145	13946
	Average for 1024 x 768		***26468***	***5634***	***531313***	***146144***
	1280 x 1024	code	17523	403	422074	7368
		desktop (many icons)	26168	677	735357	8666
		web	25964	414	758300	12538
	Average for 1280 x 1024		***23218***	***4082***	***638577***	***154539***
	1200 x 1600	code	14017	244	412181	5488
		desktop (few icons)	8877	181	388577	7076
		web	26317	140	1029823	1757
	Average for 1200 x 1600		***16404***	***7361***	***610194***	***298588***
	1280 x 1024 1200 x 1600	code	10243	356	560914	21072
		desktop (few + many)	12527	19	1060901	4107
		web	20313	141	1532615	1824
	Average for 1280 x 1024, 1200 x 1600		***14361***	***4341***	***1051477***	***399163***

Table A.4 – Average sizes of the PNG-compressed thumbnail and full view images shared.

A.3 Measuring Upload and Download Speeds

To measure typical upload and download speeds on a university and residential network, multiple websites offering speed tests were used (specifically, the top three sites on Google measuring both upload and download speeds; Table A.5, second column). For each website, upload and download speeds to various locations at different times of the day were tested on both the University of Calgary network and a residential network in a condominium. Ten speed tests were done for each condition, and overall averages were computed from the thirty speed tests done on each network (Table A.5, fourth and last row) to simplify the calculations done in Section 4.3.3.

While archives of speed test statistics for various Internet Service Providers (ISPs) are available online [e.g., www.dslreports.com/archive; www.speedtest.net/global.php], they tend to report on the fastest speeds measured for each ISP, rather than on typical speeds. The averages reported in Table A.5 are intended to provide more realistic measures of network speed that are used to give concrete examples of Screen Sharing item performance in Section 4.3.3.

Network Measured	**Website Address**	**Upload (kbps)**		**Download (kbps)**	
		Mean	**SD**	**Mean**	**SD**
university	www.dslreports.com/speedtest	3284	1614	6893	2948
	www.speakeasy.net/speedtest	5993	1644	6165	2225
	www.speedtest.net	5755	2072	6888	3050
	Average for university	***5011***	***2128***	***6649***	***2691***
residential	www.dslreports.com/speedtest	485	13	1392	281
	www.speakeasy.net/speedtest	370	127	1683	310
	www.speedtest.net	414	114	2092	1037
	Average for residential	***423***	***107***	***1723***	***310***

Table A.5 – Measured upload and download speeds for the University of Calgary network and a residential network at home.

Appendix B. Study Materials

This appendix contains documentation related to the evaluations described in Chapter 5: the internal study involving members of my own research group, conducted in March 2006; and the interviews with the external development team, done in May 2006 with Gregor McEwan. The contents of this appendix are as follows:

- ***Ethics Approval:*** Ethics approval for this research was granted by Janice Dickin, Chair of the Conjoint Faculties Research Ethics Board at the University of Calgary, in November 2004.

- ***Consent Form:*** Participants from the external development team were required to read and sign this consent form prior to their interviews.

- ***Interview Questions:*** Participants were asked some of these questions during their interviews, as well as additional follow-up questions.

UNIVERSITY OF
CALGARY

CERTIFICATION OF INSTITUTIONAL ETHICS REVIEW

This is to certify that the Conjoint Faculties Research Ethics Board at the University of Calgary has examined the following research proposal and found the proposed research involving human subjects to be in accordance with University of Calgary Guidelines and the Tri-Council Policy Statement on *"Ethical Conduct in Research Using Human Subjects"*. This form and accompanying letter constitute the Certification of Institutional Ethics Review.

File no:	**4162**
Applicant(s):	**Kimberly E. Tee** Gregor McEwan
Department:	**Computer Science**
Project Title:	**Exploratory Study of Community Bar**
Sponsor (if applicable):	

Restrictions:

This Certification is subject to the following conditions:

1. Approval is granted only for the project and purposes described in the application.
2. Any modifications to the authorized protocol must be submitted to the Chair, Conjoint Faculties Research Ethics Board for approval.
3. A progress report must be submitted 12 months from the date of this Certification, and should provide the expected completion date for the project.
4. Written notification must be sent to the Board when the project is complete or terminated.

Janice Dickin

Janice Dickin, Ph.D, LLB,
Chair
Conjoint Faculties Research Ethics Board

2004/11/08
Date:

Distribution: (1) Applicant, (2) Supervisor (if applicable), (3) Chair, Department/Faculty Research Ethics Committee, (4) Sponsor, (5) Conjoint Faculties Research Ethics Board (6) Research Services.

Gregor McEwan and Kimberly Tee
Department of Computer Science
University of Calgary
2500 University Drive
Calgary, AB, CANADA T2N 1N4

Consent Form for Participants

Research Project: Exploratory study of Community Bar

Investigators: Gregor McEwan and Kimberly Tee
Supervisor: Saul Greenberg

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information.

Description of Research Project:

The purpose of this study is to explore how people use an informal awareness and casual interaction tool, Community Bar. We will conduct an interview where we will ask a series of questions to help us evaluate the design and functionality of Community Bar in terms of how it is used. With your permission, this interview will be recorded for later analysis.

Participation in this interview will not put you at any risk or harm and is strictly voluntary. You may choose to withdraw from this interview at any time. Any data collected to your withdrawal will still be available to the investigators for analysis.

Electronic data will be stored in a secure manner, such as in a computer secured with a password. Hardcopies of data will be stored in a locked cabinet/room with restricted access. Data will be kept for a minimum of three years. On disposal, electronic data will be erased and hardcopies will be shredded.

Personally identifiable information will only be used in papers or presentations with your explicit permission. If we wish to use any personally identifiable information, we will contact you with the particulars of the information we wish to use, and you may decide whether or not you give us the permission to use it. If you choose at this time not to give us permission to use personally identifiable information we collect in papers or presentations, please indicate so below by checking the box:

□ I do **not** give the investigators permission to use personally identifiable information in papers and presentations.

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a subject. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from the interview at any time. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation.

At the conclusion of the interview and its analysis, we will post any resulting papers that we have written about them. You can view these by asking the investigator or by accessing the website: http://grouplab.cpsc.ucalgary.ca/papers/index.html

If you have further questions concerning matters related to this research, please contact:

Gregor McEwan	(403) 210-9499	mcewan@cpsc.ucalgary.ca
Kimberly Tee	(403) 210-9499	tee@cpsc.ucalgary.ca

If you have any questions or issues concerning this project that are not related to the specifics of the research, you may also contact the Research Services Office at (403) 220-3782 and ask for Mrs. Patricia Evans.

Participant's Name

Participant's Signature　　　　Date

Investigator and/or Delegate's Signature　　　　Date

Witness' Signature　　　　Date

A copy of this consent form has been given to you to keep for your records and reference.

B.3 Interview Questions

During their interviews, participants were asked some of the following questions, as well as additional follow-up questions. Participants who did not share their screen were only asked the questions applying to audience members.

As Owner (Research Group)

- What did you like about the Screen Sharing item?
- What did you dislike?
- Did you have any experiences (good or bad) that you want to tell me about?
- What settings did you use? Why?
- How come you did/didn't choose to use other types of image distortion?
- What concerns did you have?

As Audience (Research Group)

- What did you like about the Screen Sharing item?
- What did you dislike?
- Did you have any experiences (good or bad) that you want to tell me about?

As Owner (Development Team)

- What do you usually share?
- What settings do you use?
- Did you have any concerns about sharing your screen before? How have they changed?
- What do you like about sharing your screen?
- What do you dislike?
- Examples of experiences?

As Audience (Development Team)

- How well can you identify what you see on others' desktops? Examples?
- What do you like about seeing others' screens?
- What do you dislike?

General (Development Team)

- How often do you use remote pointing? Examples of when/for what you've used it for?
- What would you like to see added to the Screen Sharing item?

Appendix C. Co-Author Permissions

In this appendix, I include permissions from my collaborators to use co-authored work from our papers in my thesis.

UNIVERSITY OF
CALGARY

November 14, 2007

Department of Computer Science
University of Calgary
2500 University Dr NW
Calgary, Alberta
T2N 1N4

I, Saul Greenberg, give Kimberly Tee permission to use co-authored work from our paper and video, listed below, for Chapters 1, 2, 3, and 5 of her thesis and to have this work microfilmed.

Co-authored work:

Tee, K., Greenberg, S., and Gutwin, C. Providing artifact awareness to a distributed group through screen sharing. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2006). ACM Press, 99-108.

Tee, K., Greenberg, S., Gutwin, C., and McEwan, G. Shared desktop media item: The video. In Video Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2006). ACM Press, November, Duration 4:00.

Sincerely,

Saul Greenberg

UNIVERSITY OF
CALGARY

November 14, 2007

Department of Computer Science
University of Saskatchewan
110 Science Place
Saskatoon, Saskatchewan
S7N 5C9

I, Carl Gutwin, give Kimberly Tee permission to use co-authored work from our paper and video, listed below, for Chapters 1, 2, 3, and 5 of her thesis and to have this work microfilmed.

Co-authored work:

Tee, K., Greenberg, S., and Gutwin, C. Providing artifact awareness to a distributed group through screen sharing. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2006). ACM Press, 99-108.

Tee, K., Greenberg, S., Gutwin, C., and McEwan, G. Shared desktop media item: The video. In Video Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2006). ACM Press, November, Duration 4:00.

Sincerely,

Carl Gutwin

Carl Gutwin

UNIVERSITY OF
CALGARY

November 14, 2007

Department of Computer Science
University of Calgary
2500 University Dr NW
Calgary, Alberta
T2N 1N4

I, Gregor McEwan, give Kimberly Tee permission to use co-authored work from our video, listed below, for Chapters 1 and 3 of her thesis and to have this work microfilmed.

Co-authored work:

Tee, K., Greenberg, S., Gutwin, C., and McEwan, G. Shared desktop media item: The video. In Video Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2006). ACM Press, November, Duration 4:00.

Sincerely,

Gregor McEwan