Understanding and Mitigating Display and Presence Disparity in Mixed Presence Groupware

Anthony Tang, Michael Boyle and Saul Greenberg

Department of Computer Science
University of Calgary
2500 University Drive N.W.
Calgary, Alberta, Canada T2N 1N4
Email: tonyt@cs.ucalgary.ca, boylem@cpsc.ucalgary.ca, saul@cpsc.ucalgary.ca

Mixed Presence Groupware (MPG) supports both co-located and distributed participants working over a shared visual workspace. It does so by connecting multiple single-display groupware workspaces together through a shared data structure. Our implementation and observations of MPG systems exposes two problems: the first is display disparity, where connecting heterogeneous displays introduces issues in how people are seated around the workspace and how workspace artifacts are oriented; the second problem is presence disparity, where the perceived presence of collaborators is markedly different depending on whether they are co-located or remote. Presence disparity is likely caused by inadequate consequential communication between remote participants, which in turn disrupts group collaborative and communication dynamics. To mitigate display and presence disparity problems, we determine virtual seating positions and replace conventional telepointers with digital arm shadows that extend from a person's side of the table to their pointer location.

ACM Classification: H.5.3 (Groups and organizational interfaces – Computer supported cooperative work).

1. INTRODUCTION

The time/space taxonomy of groupware (Figure 1) categorises applications based on where and when collaborators use them (Baecker, Grudin, Buxton and Greenberg, 1995). This taxonomy partitions groupware into four quadrants based on style of use, and work practices:

- *same time – same place* systems supporting face-to-face interactions,
- *same time – different place* systems supporting real time distributed interactions,
- *different time – different place* systems supporting asynchronous work, and
- *different time – same place* systems supporting co-located, on-going tasks.

Many applications have been designed to fit within a quadrant. Multi-Device Multi-User Multi-Editor (MMM), for example, cleanly fits within the same time – same place cell because it supports co-located collaborators sharing a single display using multiple mice (Bier and Freeman, 1991). However, this quadrant view of groupware is limiting (Baecker, 1993); in practice, people's collaborative practices cross the boundaries laid out by the taxonomy. For instance, the room

Manuscript received: 26 April 2004
Communicating Editor: Sidney A. Morris

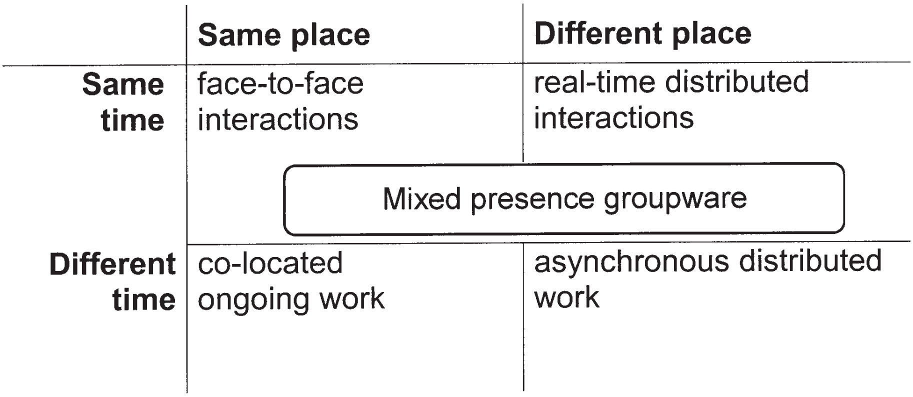

Figure 1: Mixed presence groupware in the space–time groupware matrix

metaphor in TeamWave Workplace recognizes that people's collaboration with others may span the time boundary (Greenberg and Roseman, 2003). Consequently, as multiple people enter a virtual room, they can interact synchronously over all items within a room; yet, they can also leave items in a room for absent people to work on later, thus permitting asynchronous interaction.

In the same vein, *mixed presence groupware* (MPG) supports synchronous work in a shared visual workspace by both co-located and distributed collaborators, thereby spanning the same place – different place quadrants at the top of Figure 1. Figure 2 gives an illustrative example, where the photos show several distributed groups of co-located people working over various physical displays containing a common shared visual workspace. As seen in the figure, the physical display may be a horizontal tabletop display, a vertical presentation display, or even a conventional monitor. All participants have their own input devices, and all can interact with the workspace at the same time with their actions being immediately reflected on all displays. Conceptually, the physical tables embody a virtual table surrounded by co-present and remote participants (Figure 2, bottom right).

Our own interests are in the human, social and technical factors that arise in the design and use of these MPG applications by co-located and remote collaborators. In particular, our initial implementation and observations of the use of an MPG prototype revealed two problems as summarized below and explained further in later sections.

1. ***Display disparity.*** Connecting heterogeneous tabletop and vertical displays introduces issues in how people should be positioned around the virtual table, and as a consequence how workspace artifacts would be oriented towards them. For example, consider participants 1 and 2 working opposite one another on a table display, and a connected participant 3 working at a monitor. The virtual table could distribute all participants around it on separate sides, or it could have participant 3 seated on the same side as participant 1. In either case, items drawn by participant 2 in his orientation will not appear "right-side up" for participant 3; while mis-oriented items are acceptable on table top displays (because people often stand and interact on different sides), it looks decidedly odd when seen on a vertical monitor (because we don't expect people to be atop the monitor facing down).

2. ***Presence disparity.*** A participant's perception of the presence of a fellow collaborator differs depending on whether that collaborator is physically co-located or remote. This disparity

Figure 2: Three teams working in a conceptual MPG setting over three connected displays, stylized as a virtual table in the bottom right

disrupts group collaborative and communication dynamics. We suggest that one of its causes is that consequential communication (i.e. visibility of another's body) between remote participants is inadequate.

In this article, we discuss our initial experiences in designing and building mixed presence groupware prototypes, and how we mitigate the display and presence disparity problems. We begin by situating mixed presence groupware within current groupware research efforts. We next describe the iterative design and implementation of a prototype MPG application called MPGSketch. From our experiences in building MPGSketch, and our observations of its use, we discuss the human and technical aspects of presence and display disparity. Finally, we discuss techniques for linking heterogeneous displays, and introduce digital arm shadows as a method to restore presence parity.

2. RELATED WORK ON SHARED VISUAL WORKSPACES

A shared visual workspace is one where participants can create, see, share and manipulate artifacts within a bounded space. Real world examples include whiteboards and tabletops, which allow groups to collaborate using tools like markers, paper, tape, and scissors. Electronic counterparts to shared workspaces have been developed as distributed groupware, single display groupware, and to a much lesser extent, mixed presence groupware. These environments also provide digital counterparts to the physical tools of the real world visual workspaces.

Distributed Groupware

Distributed groupware applications for shared visual displays abound, and have been a major focus for computer-supported cooperative work (CSCW) research over the past twenty years. These applications make interactions between distance-separated collaborators possible, and are attractive because they potentially reduce travel time and costs associated with remote collaboration. For example, globally-minded enterprises are trying to use distributed groupware tools to assemble agile, cohesive and productive teams out of workers located in different countries (Rogers, 1994). Yet the design of these tools is fraught with social and technical challenges whose solutions are non-obvious. A large body of theoretical and empirical knowledge about these challenges has emerged from CSCW research in distributed groupware (e.g. Baecker, 1993; Gutwin and Greenberg, 2002), and several toolkits are now available to assist the researcher in rapidly prototyping distributed workspaces (Greenberg and Roseman, 1999).

Single Display Groupware

While distributed interaction is clearly important, the bulk of a person's day-to-day interactions are with co-located individuals. This fact motivated research into computer systems to better support the affordances of co-located collaboration. Single display groupware (SDG) challenges the conventional 1:1 ratio between users and computers by allowing multiple users, each with his/her own input device (e.g. a mouse, a stylus, etc.), to interact over a shared display simultaneously (Stewart, Bederson and Druin, 1999). Early experiences with SDG systems indicate that they support natural dynamics of collaboration and conversation better than distributed groupware. Yet designing usable SDG interfaces and interactions remains difficult: hard technical factors include getting multiple devices to appear as independent input streams (Tse and Greenberg, 2002); hard social factors include recognizing the roles of workspace artifact orientation and personal space in mediating activity (Kruger, Carpendale, Scott and Greenberg, 2002). Although many important factors have yet to be thoroughly investigated, research into SDG has advanced to the point where toolkits for rapidly prototyping these systems are now available (e.g. Tse and Greenberg, 2002).

Mixed Presence Groupware

Given this research on both distributed and single display groupware, one would expect equivalent advances in groupware that merges these concepts into MPG. Surprising, very few examples of this type of groupware exist in the literature. One is the Touch Desktop, created as part of the Swedish Institute of Computer Science's investigation into natural interaction within multi-user CAVE-like environments (Hansson, Wallberg and Simsarian, 1997). Co-located people work on a touch screen tabletop display, which is placed in front of the "communications wall" containing the 3D virtual environment (Figure 3). Actions on the physical table are reflected on the graphical table located in the virtual environment; thus remote collaborators, represented as avatars in the virtual environment, can see what others are doing. However, the authors provide little additional information, and we suspect the system does not incorporate multiple physical tables.

A commercial example of MPG is Halo, a multi-player game for Microsoft's Xbox. Co-located players can interact through a split-screen, and distributed groups of players can be connected together by connecting several Xboxes together. All players and their actions are visible in each person's scene.

Perhaps the most common examples of MPG are based on video conferencing technology. These systems typically use a video channel that transmits the image of co-located participants' work over a drawing surface, or special overlays are used allowing people to annotate over this video image. Some research systems even provide a shareable video-based drawing area by over-

Figure 3: Touch Desktop. Photo from Swedish Institute of Computer Science (Photo from http://www.sics.se/~par/dive_docs/interaction.html).

laying the images of multiple video cameras (e.g. Tang and Minneman, 1991a; Tang and Minneman, 1991b; Ishii and Kobayashi, 1992). While demonstrations of these systems typically show these systems as a means for connecting two distributed collaborators, multiple co-located participants can be supported by simply moving them into the scene. An unfortunate constraint of these systems is that participants cannot alter artifacts on the drawing surface introduced by remote participants.

Finally, people often work in an MPG-like mode even when the software does not support it. For instance, instant messengers explicitly support only one user per terminal chatting with others on their own terminals. However, others may chat "over the shoulder" by telling the co-located partner what to type, or by taking control of the mouse and keyboard.

Our focus on MPG is distinct from this prior work. First, we are interested in supporting scenarios that allow multiple co-located teams to gain equal access to a single shared drawing surface. Second, all participants have their own input device, where each can manipulate the shared space–even simultaneously–at any time.

3. MPGSKETCH: A MIXED PRESENCE GROUPWARE DRAWING SYSTEM

Our first goal was to understand the technical challenges of building MPG applications, and to gain some initial experiences in using one.

3.1 Description

We began our investigation by implementing and using MPGSketch, a simple MPG real-time shared drawing application that collects distance-separated groups of co-located collaborators in a shared visual workspace. Participants sketch over an empty surface, an image taken from a file, a snapshot captured from a webcam, or a screen-grab of a desktop. A screen capture of MPGSketch is shown in Figure 4, while Figure 2 shows participants sketching over this image.

Each person has his or her own pointing device for input (e.g., finger on touch-sensitive table, pen on vertical whiteboard, or mouse). Multiple cursors, labeled with their owners' names, show the location and movement of the pointing devices in the workspace. The shared workspace presents these cursors and the drawing as it evolves. Any participant can draw on the display at any time, where their drawing actions can occur simultaneously. These drawing actions are reflected

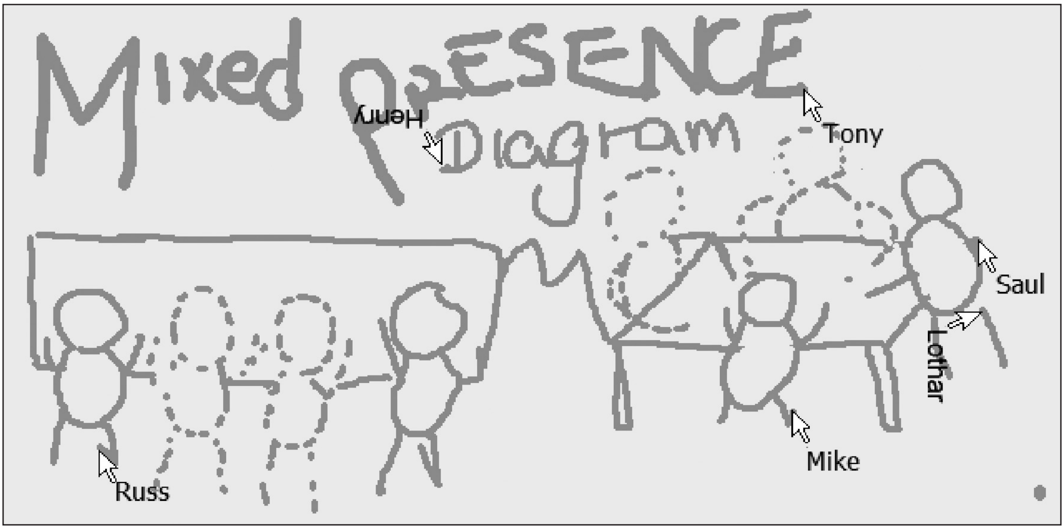

Figure 4: MPGSketch with six participants, each with a telepointer that reflects his or her position in the workspace

immediately on all displays. What makes MPGSketch an MPG application is that, as illustrated in Figure 2, several individuals can work on a single display, and that this display is connected to remote displays being worked on by others.

3.2 Implementation

Because MPG applications are rare, it is worth taking a moment to describe the infrastructure and design of MPGSketch.

We had two groupware toolkits at our disposal, both developed in our laboratory: the Single Display Groupware Toolkit (SDGToolkit) (Tse and Greenberg, 2002), and the GroupLab Collabrary (Boyle and Greenberg, 2002). Each of these toolkits provides prototyping capabilities for different kinds of groupware. Together, the toolkits provided the necessary infrastructure for MPGSketch.

The Single Display Groupware Toolkit (SDGToolkit) is a toolkit for rapidly prototyping SDG. It automatically captures and manages multiple mice and keyboards, and presents them to the programmer as uniquely identified input events relative to the application window. It transparently provides multiple cursors and text captions, one for each mouse. To handle orientation issues for tabletop displays, the SDGToolkit automatically rotates the cursor and translates input coordinates so the mouse behaves correctly. The SDGToolkit also provides an SDG-aware widget class layer that significantly eases how programmers create novel graphical components that recognize and respond to multiple inputs.

The SDGToolkit makes it easy to convert conventional single user applications into applications that support multiple users. It is integrated into Microsoft Visual Studio as a graphical component that can be easily dropped into any existing application. Programmers then create an SDG application using familiar programming constructs such as object properties and methods, events and callbacks. Finally, although the SDGToolkit was originally designed for multiple mice and keyboards, it also supports several input devices for large wall or table displays. In sum, the SDGToolkit makes it easy to create single display groupware applications; however, it does not provide support for distributed participants.

The GroupLab Collabrary (Boyle and Greenberg, 2002) is a toolkit that combines easy access to audio and video capture and manipulation with a groupware application data sharing infrastructure. Although it is specifically designed to aid rapid prototyping of media spaces (Bly, Harrison, and Irwin, 1993), the Collabrary's simple and generic API makes it easy to consume all or part of its functionality in other kinds of applications. In our digital arm shadows application, we make use of the groupware data sharing, but not the multimedia facilities.

The data sharing functionality of the Collabrary is made available through its *shared dictionary* component. The shared dictionary is a fully-replicated data structure that maps string keys to values of any network-marshallable type (e.g., integers, strings, arrays, bitmaps, and even complex programmer-defined objects). The string keys mimic paths in a typical UNIX-like file system and a simple pattern-matching language, akin to that used with file systems, permits hierarchical grouping of dictionary entries. As a result, programmers gain the benefits of aggregation and encapsulation of a hierarchical data structure with the ease of access of hash table. In fact, storing or retrieving a value in the shared dictionary is as straightforward for the end-programmer as is accessing a value in an array, and is accomplished in a syntactically identical manner.

Updates to the dictionary are serialized through a centralized server architecture and pushed out to clients using a proprietary binary network wire protocol over persistent TCP connections (Boyle, 2003). Thus, the shared dictionary may be considered a notification server (Ramduny, Dix, and Rodden, 1998) with cached keys and values. No separate server software is used: the server and client are implemented in the same object library and servers may be started programmatically just as easily as clients may be connected. End-programmers may *subscribe* to changes to patterns of keys in the dictionary. With these subscriptions, they receive programmatic notification of additions, modifications, and deletions to entries in the dictionary. The end-programmer can attach a notification event handler to, say, update a GUI with new information stored in the dictionary. With these subscription notifications, the shared dictionary permits distributed groupware development using the model-view-controller paradigm used in the GroupKit toolkit (Roseman and Greenberg, 1996).

We merged the capabilities of these toolkits to build MPGSketch (Figure 5). The SDGToolkit manages the multiple mice/keyboards attached to each computer, and draws the local cursors. It

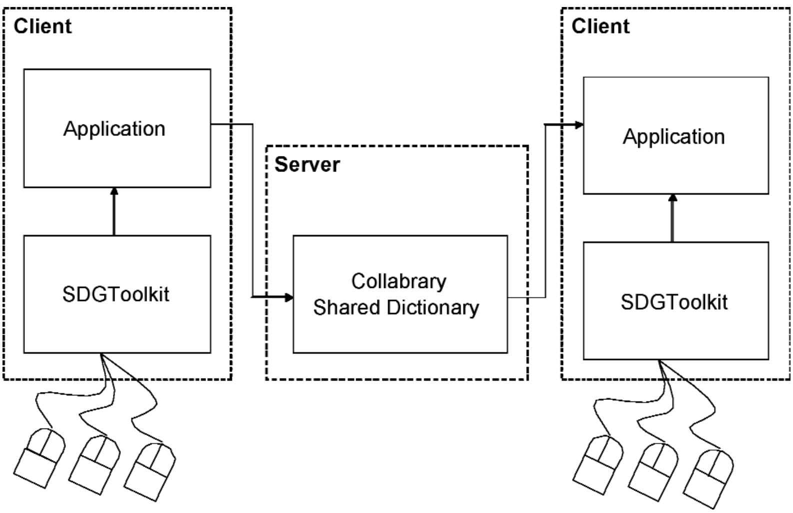

Figure 5: Architectural diagram for MPGSketch

assigns each mouse a globally unique identifier and tracks the coordinates of its corresponding cursor. The MPGSketch instance then distributes this data via the Collabrary shared dictionary to other MPGSketch instances running on different computers. It stores mouse identifiers and updates the cursors' on-screen coordinates as they move. Remote MPGSketch instances (using the cursor component of the SDGToolkit) then draw cursors at the correct location for all of the remote input devices listed in the shared dictionary. Finally, as someone draws, the drawing coordinates are also placed in the shared dictionary. Based on this information, the MPGSketch instances update the drawing to give the shared view.

In principle, this approach for prototyping mixed presence groupware applications is generalisable. While our version depends on the SDGToolkit and the GroupLab Collabrary, other tools with similar capabilities would suffice.

4. DISPLAY DISPARITY IN HETEROGENEOUS DISPLAYS

Single display groupware tends to focus on large displays (whiteboard or tabletop sized) because of their inherent ability to support co-located groupware. Because we envision MPG to be a natural extension of single display groupware, we began by considering a very plausible scenario of MPG use, where one site uses a large tabletop display while another uses a large whiteboard display. As we will see, connecting such heterogeneous displays leads to *display disparity* that, in turn, introduces a number of issues:

- How does the system know where users are sitting around the horizontal display?
- How do we mechanically and visually orient pointing devices (e.g. mice) to reflect a participant's seating position? How should this orientation be treated on local displays and remote displays?
- How do we manage "non-upright" orientations on upright displays?
- How do we manage "non-upright" orientations on remote horizontal displays?

The display disparity problems arise because, unlike monitors and other vertical displays, tabletops have sides and lack an absolute notion of up and down. The notion of which side is "up" is either undefined or arbitrary. Given this uncertainty, what does it mean to work around a table, and what does it mean to connect vertical monitors and horizontal tables?

Tabletop Orientation

Unlike vertical displays, people can be seated across or at right angles from one another around tabletop displays. Multiple seating arrangements introduce mechanical and visual orientation issues (Kruger, Carpendale, Scott and Greenberg, 2003). Suppose North is the traditional upright location. First, people in a non-North seat will be holding their mice at non-upright angles, which means that coordinates being returned by the devices are incorrect. Second, content (including labeled cursors) oriented correctly for one person will appear sideways or upside down to others. This problem is not particular to MPG – rather, the problem applies more generally to tabletop single display groupware.

Fortunately, the SDGToolkit recognizes tabletop orientation. Each mouse can be associated with a side of the table (and implicitly, an orientation): North, South, East and West. All internal mouse coordinates are transformed relative to that orientation, so that the mouse behaves correctly for the user. Similarly, the labeled cursor is automatically oriented with respect to that orientation. However, the toolkit does not enforce any strategy for content orientation or reorientation – these remain up to the implementer and are open research questions (Kruger, Carpendale, Tang and Scott, 2004).

Heterogeneous Orientation

Though much research is focused on strategies for managing object orientation on a single tabletop display, it does not solve the MPG-specific *display disparity* problems of what to do when multiple heterogeneous displays, including tabletop and vertical displays, are connected.

What does it mean to connect vertical monitors with horizontal tabletops? One problem is that we need to establish their relative orientations. As a simplistic solution, we can assume that vertical monitors are always oriented in the North position; we can then arbitrarily assign a table a North position and demand that people work side by side at that position. However, this solution results in “overcrowding” of the North side (see Figure 2, bottom right). Another approach distributes several collaborators around the table, but this marginalizes collaborators at non-North orientations.

Even if we do assign North side to the vertical display, we are left with the problem of how to display other non-upright orientations. For instance, South’s cursors and actions will be upside-down, while East and West’s actions will be sideways (e.g., see Henry and Lothar’s cursors in Figure 4). While this is expected over tabletop displays, it looks decidedly odd – even unsettling – when this happens on a vertical display. We could translate cursors so they at least appeared right-side up on the vertical display, but this solution would not work for items drawn on the surface that retain their orientation (e.g. text); furthermore, it would be misleading to remote collaborators (cursor orientation implies a collaborator’s seating orientation).

If we do not fix orientation, another problem is how people choose “sides” of the virtual work surface. With joined tabletop displays, we need to at least determine which side is North. With vertical displays, we need to specify what side of the virtual table corresponds to the bottom of the display. One strategy is to let people do this manually. Another strategy is to have the system assign sides (e.g. to prevent overcrowding of any one side, it may try to balance people around the sides of the virtual work surface). Alternatively, as in the case with all vertical displays, the system may favour a single side to give the majority a common orientation.

5. EMBODIMENT AND PRESENCE DISPARITY IN MPG

Next, we conducted an informal exploratory study of how two distributed groups would use MPGSketch. To temporarily finesse the orientation issue, we used only upright monitors with common “North” orientation. We placed two pairs of participants (all knew each other well) in front of conventional workstation monitors on either side of a partition. Each workstation ran an instance of MPGSketch and had two attached mice. While people on one side of the partition could not see those on the other side, they could clearly hear them as they spoke. The four people then performed a non-competitive collaborative sketch. While this experimental situation appears suspect – numbers are small and the task is uncontrolled – it was appropriate for our first foray into MPG use. We were looking for “big effects” – obvious issues, failures and successes – to guide our future investigations, and as typical in early testing, these are often seen in even very limited study situations.

All people were able to draw, and we saw no immediately obvious problems associated with group drawing. This success is likely because we derived MPGSketch’s design from a rich literature of observations of how people draw together (Tang, 1991), and from our own experiences of similar systems supporting either remote or co-located drawing.

However, we were surprised to observe that most of the participants’ spoken utterances were directed toward their co-located partners. Rarely, if at all, did participants speak across the partition to the remote group. That is, there was a *conversational disparity* between co-located and remote participants. This disruption to natural conversational dynamics is clearly a major issue, as disruptions to conversational dynamics necessarily disrupt collaborative dynamics. To understand why conversational disparity occurred, we looked into the role of people's *embodiments* and the differences in *presence* they introduce in co-located – distributed real-time work.

5.1 Embodiments in the physical world

A person's body interacting with a physical workspace is a complex information source with many degrees of freedom. Bodily actions such as position, posture, and movements of head, arms, hands and eyes unintentionally "give off" information which is picked up by others (Baker, Greenberg and Gutwin, 2001). These actions are a source of information, called *consequential communication* (Segal, 1995), for other co-located people since "watching other people work is a primary mechanism for gathering awareness information about what's going on, who is in the workspace, where they are, and what they are doing" (Gutwin, 1997). Unintentional body language can be divided into two categories, as described below (Baker *et al*, 2001).

Actions coupled with the workspace include gaze awareness (i.e. knowing where another person is looking), seeing a participant move towards an object or artifact, and hearing characteristic sounds as people go about their activities in the workspace. These actions inform others of many things. First, one's proximity to the workspace indicates: whether one can see the contents of the workspace, one's ability to reach into the workspace, and one's relative orientation to workspace artifacts. Second, body and hand motions tend to be large and take time to do, allowing others to infer and react to that person's intentions. For example, when others see a person's hand move over the drawing surface toward a pair of scissors, they can anticipate what that person is about to do. Others can modify their own actions accordingly, for example, to avoid conflict, to support the person's actions, or to repair potential problems before they occur.

Actions coupled to conversation are subtle cues picked up from our conversational partners that help us continually adjust our verbal behaviour (e.g. Clark, 1996). Some of these cues are visual ones coming from a person's embodiment: facial expressions, body language (e.g. head nods), eye contact, or gestures emphasizing speech. These visual cues provide *conversational awareness* that helps people nurture conversation. These cues allow people to mediate turn-taking, focus attention, detect and repair conversational breakdown, and build a common ground of joint knowledge and activities (Clark, 1996). For example, eye contact helps determine attention: people will start an utterance, wait until a listener begins to make eye contact, and then start the utterance over again (Goodwin, 1981). On a coarser level, the proximity of one's body to another suggests different degrees of presence. Presence is important since it is an essential cue used in initiating, continuing, and terminating conversation (Lombard and Ditton, 1997). Many informal awareness cues for presence are visual in nature; for instance, people who are physically close are visually much larger than people who are far away. The visually large embodiments of co-located collaborators (compared to the telepointer embodiments of remote collaborators) make co-located collaborators appear comparatively more present.

While the above discussion deals with consequential communication, a person's embodiment also plays a significant role in *intentional communication*. These include explicit gestures and other visual actions used along verbal exchanges. For example, Tang (1991) observed that gestures play a prominent role in all work surface activity for design teams collaborating over paper on tabletops and whiteboards (around 35% of all actions). These are intentional gestures, where people used them to directly support the conversation and convey task information. Intentional gestural communication takes many forms (Baker *et al*, 2001). *Illustration* occurs when speech is illustrated, acted out, or emphasized. For example, people often illustrate distances by showing a gap between

Figure 6: Corporeal arms in a common workspace

their hands. *Emblems* occur when words are replaced by actions, such as a nod or shake of the head indicating “yes” or “no” (Short, Williams and Christie, 1976). Deictic reference or deixis happens when people reference objects in the workspace with a combination of intentional gestures and speech, e.g. by pointing to an object and saying “this one” (Clark, 1996).

Figure 6 brings these concepts to life. While we only see the arms on the workspace in this cropped photo, we immediately notice that two people are present, that both are poised to do work over specific places in different documents (by the position of the pens), and that the person on the right is pointing at an image with her pen, emphasizing the image with the other hand. The arm postures signal that both are engaged in the conversation.

5.2 Embodiments in MPGSketch

As with many real-time groupware systems, MPGSketch represents all remote participants with cursors (or telepointers), as seen in Figure 4. In distributed groupware, this small cursor (typically 32x32 pixels) is a remote user’s only embodiment in the shared workspace when they are not actively drawing. While cursors are simple, they have proven effective in distributed settings. The presence and movement of the cursor serves as the visual representation of a remote person’s presence and activity, and people show remarkable resilience against the missing information, often altering work and conversational strategies.

The problem in mixed presence groupware is that there is a huge disparity between the embodiments of remote people (cursors), and the real-world embodiments of local people (bodies). We call this difference *presence disparity*. For example, contrast people’s real world arm embodiments in Figure 6 with the cursor embodiments in Figure 4. The size disparity alone is a major factor: arms are many orders of magnitude larger than a remote user’s cursor, and thus commands much more attention. The low information richness and accuracy of the cursor embodiment is another factor. For example:

- Cursors may suggest where its owner is looking but cannot guarantee it.
- An idle cursor (i.e. one that remains stationary for a while) suggests a person's presence, but again cannot guarantee it.
- The orientation of a cursor suggests where its owner is seated at a virtual table, but cannot indicate how the person may actually be seated relative to that display in real life.
- Cursor gestures are reduced to deixis, with emblems and illustrations difficult to perform.
- Cursors cannot transmit bodily proximity to others (e.g. as happens in real life when a person leans in towards another to initiate conversation).
- While people normally initiate computer actions with their mouse, some cursor actions may be too quick or even invisible for others to see. This interferes with others' ability to infer intentions, and to react to them in a timely manner.

We believe that the presence disparity caused by the embodiment differences lead to the conversational disparity seen in mixed presence groupware. Because co-located embodiments dominate in presence through their size and richness, people direct nearly all of their utterances to co-located collaborators.

6. REBALANCING DISPLAY AND PRESENCE DISPARITY WITH DIGITAL ARM SHADOWS

In the second iteration of our MPG prototype, we refocused our efforts to manage seating issues, and to provide remote users with better embodiments.

6.1 Seating Rules

Traditional groupware applications connect several upright displays together. The orientation of the shared workspace on these displays is identical – it would be odd to consider anything but "North" orientation in these scenarios. In connecting upright and tabletop displays, display disparity means that some users at horizontal displays will invariably be at non-default (non-North) orientations. Without special treatment, the model of the shared workspace and its participants would be as represented in Figure 2, lower right – a vast majority of users (those who are using upright displays) sitting at one side of the table with a given orientation, and a minority of users (a subset of those using horizontal displays) sitting at different sides of the table – each with a different orientation. While we do not know if this overcrowding is good or bad, we do believe that a few reasonable heuristics can help distribute participants around the sides of the virtual table while preserving the physical orientation of co-located users.

1. Users' locations around physical tables are preserved around the virtual table.
2. Users who are seated side by side at an upright display remain seated next to one another at the virtual table.
3. Connected upright displays are automatically placed at different sides of the table.

6.2 Detecting User Presence

An interesting problem is how the system detects a user's presence. Presenting an embodiment for each enumerated input device is misleading for remote participants, because there may not actually be that many users in the workspace. Specifying the number of users when starting the application is too restrictive – participants in the workspace may arrive late, which would require restarting the application. Instead, we designed two implicit mechanisms to detect user presence. First, we recognize when a person sits on a particular chair around the table by embedding a light sensor in its seat. Thus, when the light sensor goes dark, we know someone has sat down (sensors are implemented via Phidgets: Greenberg and Fitchett, 2001). Of course, this solution requires fixed seating – since a seat is implicitly bound to some input device, moving seats around the table would require system recalibration.

Thus, our second mechanism for detecting user presence monitors the mice movements, where each mouse is assigned to a particular seat. When people first sit down at a computer, they often wiggle their mouse rapidly to find their mouse pointer on-screen. We see this action as an informal way of greeting the computer – a presence signal. We detect the absence of the user through an inactivity timeout.

These two binary approaches to presence are somewhat simplistic and are prone to error. Furthermore, a fairly large literature exists that conceives of presence as a deeper notion with many facets (for a review, see Lombard and Ditton, 2001). For instance, do we present embodiments for lurkers – those who watch, but do not actively participate in the collaboration on the workspace? If not, their presence may have an effect on co-located users, but not on remote users. While presence may be a larger, more complex construct, we believe our approaches will work well in practice for most display scenarios.

With these methods of detecting presence in hand, we now discuss the digital arm shadows as a method for representing and presenting presence information.

6.3 Digital Arm Shadows as Indicators of Social Presence

Once participants are seated, we now need to communicate the orientation of each participant to others. For inspiration, we turned to VideoWhiteboard (Tang and Minneman, 1991b), a video-based tool that provides a large shared drawing area between two sites. In VideoWhiteboard, video cameras behind translucent drawing surfaces capture all activities on and near each surface, including not only the marks made on the surface with a felt pen, but the shadow of the users' bodies (usually hands and arms) as they move atop it. The video from both sides are then fused, creating a composited image, partially creating the scene in Figure 4. Thus, the user's arm gracefully appears as a shadow on the workspace as he moves toward the workspace, and disappears as he moves away from it. These arms are not only visually large, they are also socially natural indicators of presence. While extremely effective, VideoWhiteboard has technical limitations. It has high setup and equipment costs, people cannot edit each others' marks, and it does not scale well because image degradation increases with the number of overlayed video streams.

Although tabletop and upright displays are not the same as whiteboards, we thought that arms might also make suitable embodiments in our MPG prototype. Thus, we created *digital arm shadows* for remote collaborators that incorporate the properties of presence as in Video-Whiteboard. Using real arms working over tables as our model (such as Figure 6), each arm shadow maintains a 135° articulation, and roughly maintains natural forearm/upper-arm and width/length proportions. The "shoulder" point of the arm is attached to one of the sides of the table, and the "hand" point is bound to the mouse cursor location. The shadows themselves are semi-transparent, allowing objects on the underlying workspace to show through.

The arm shadows are packaged as an independent software component (i.e. a widget) that can be incorporated into MPG applications. Through a simple programmatic interface, the programmer can bind the hand of the digital arms to telepointer locations, and the shoulder point to several positions around the display.

MPGSketch's telepointers were replaced with arm shadows to represent participants. Figure 7 gives an example, with two peope at the East and West sides of a large display, and one person at the North side of a table. To show presence and absence, the shadow appears when a user's presence

Figure 7: Presence with digital arm shadows

is detected, and disappears when the user leaves. For example, when a person sits down at a chair, or begins using the mouse, the system conveys this presence information to all clients by drawing a corresponding arm shadow for that user. The system also conveys its uncertainty of one's actual presence by slowly increasing the transparency of the digital arm shadow when the owner has been inactive. The software embodiment thus has a property of a real-life embodiment: the embodiment is only present when the person is physically present and active over the surface. In contrast to most other groupware systems, our system recognizes presence as an individual's physical presence and activity at the terminal as opposed to a software client's connection to the server.

Although not implemented, we brainstormed an idea that could potentially link live video portraits of participants with their respective shoulders (Figure 8). In each case, we would capture live video streams that would be background subtracted, resulting in small portraits of each collaborator. Pinning these portraits to the collaborator increases identity, and allows other body language to come through the video; however, it does compromise space on the display.

To summarize, our contention is that arm shadows trigger the belief of remote collaborators' presence by reproducing several key attributes of real-life embodiments (as in Figure 6) above and beyond those offered by standard telepointers:

- ***Indicates virtual seating position.*** Digital arm shadows appear from a side of the application window frame much as a person's corporeal arms appear from a person's seating position, thereby "grounding" the virtual arms to an imagined virtual body.
- ***Conveys person-specific orientation.*** Each arm has a different orientation, fostering the impression that each user has a distinct view of the display. By conveying distinct, person-specific orientations, drawings by each participant can be interpreted correctly (Kruger *et al*, 2003). For instance, Tang (1991) noticed that drawings oriented toward its creator tend to be personal, while those oriented towards others tend to be public.
- ***Increased awareness of actions.*** A participant's actions are far more visible to others when compared to telepointers. First, our transluscent shadows partially obscure the workspace underneath the arms, just as real arms obscure part of the table (Figure 6). Second, digital arm shadows are large (about an order of magnitude larger than telepointers), thereby commanding more attention.

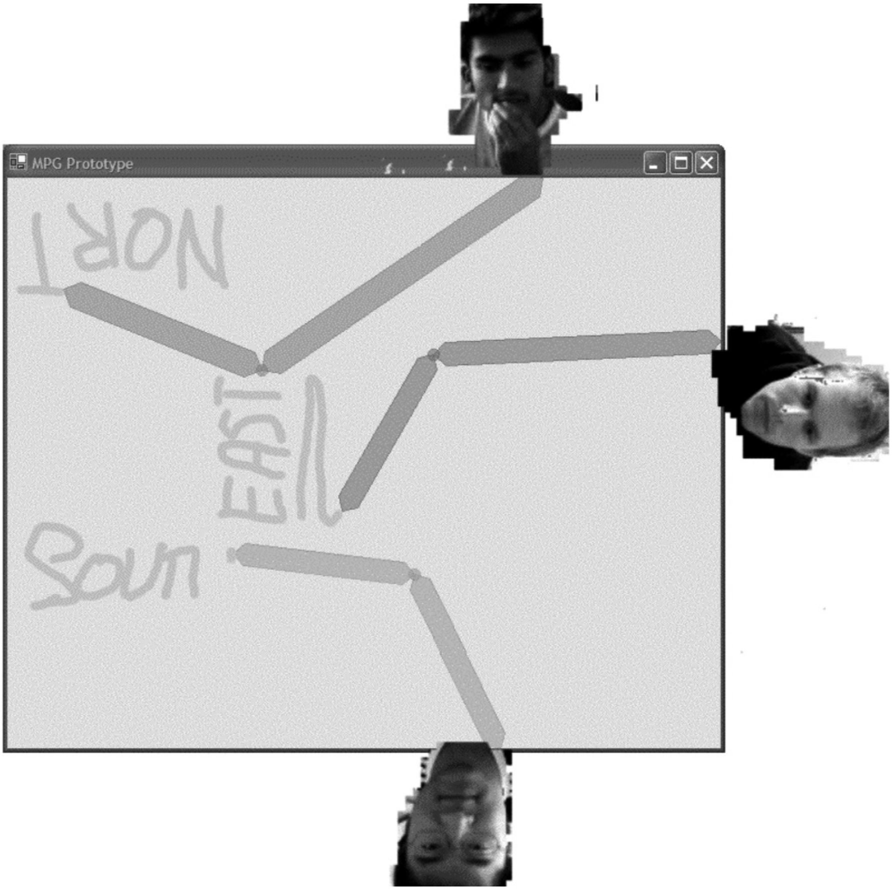

Figure 8: Enhancing presence through live video portraits

- ***Transmits identity.*** People have extremely varied physical appearances – body/face size, shape and proportion, skin colour, hair, clothes, etc. – that are the essential cues for identity. Although our arms are far from photorealistic, they can be customized to approximate real arms and thus unambiguously represent other users. Current customizable arm parameters include colour and proportion. Adding video portraits (Figure 8) can increase identity substantially, at the cost of screen space.

These properties of digital arm shadows, taken together, are virtualizations of real-life properties found in corporeal arms above and beyond those offered by standard telepointers.

7. DISCUSSION AND SUMMARY

The presence and display disparity problems we have discussed in this article are particular to mixed presence groupware systems. While the prototype MPG application presented is an example of an MPG shared visual workspace, acquiring and representing presence information appropriately is a general problem applicable to a wide array of distributed groupware systems. For example, signaling presence is an essential function of instant messaging systems (Nardi, Whittaker and

Bradner, 2000). Also, collaborative virtual environments (e.g. Benford, Greenlagh, Bowers, Snowdon and Fahlén, 1995) and media spaces (Gaver, Moran, MacLean, Lövestrand, Dourish, Carter and Buston, 1992) all seek to provide rich, socially natural embodiments for presence and informal awareness because, as suggested earlier, presence plays a vital role for regulating conversation. In tele-presentation and videoconferencing for distributed learning, local and remote audience members interact through video and audio links. Presence disparity in particular could negatively affect the learning experiences of students who must rely on the mediated link for interactions with their teachers. The TELEP system (Jancke, Gruden and Gupta, 2000) for example, provided remote audience members with embodiments in a lecture theatre so that speakers could better field questions from remote viewers.

Our focus in this article was on dual co-located/distributed synchronous groupware, which we called mixed presence groupware (MPG). To help us understand design issues in this new class of groupware, we developed a prototype MPG groupware application which we hoped would afford users the benefits of both remote collaboration from distributed groupware and of increased social interactions from single-display groupware. Instead, we saw that most of our users' utterances were directed toward their co-located partners. We attributed this social dynamic to presence disparity: the presence of remote collaborators is weakly perceived relative to co-located collaborators. We believe that this diminished sense of presence impairs normal conversational dynamics. We also saw orientation problems arising from differences between display types and how people were seated around the virtual table, which we called display disparity.

We adapted our prototype to work with heterogeneous display scenarios, where the system handled participant seating and orientation. To the prototype we added digital arm shadows as a rich embodiment for presence. We chose digital arm shadows because they offered a variety of rich properties that we believe are important to signaling presence. We believe that another person's physical presence triggers a set of mental processes that regulate social dynamics; our aim is to distill the numerous properties of physical presence to an essential subset required to trigger these mental processes – this false belief of remote collaborators' presence.

Of course, these are early experiences in MPG. We have identified two critical factors – display and presence disparity – but there are likely others in MPG design. While we have demonstrated several "solutions" to these issues, they are best considered as design explorations rather than recommended practice.

ACKNOWLEDGEMENTS

Thanks to Edward Tse for his technical expertise and his work on drafts of this article. Thanks to the members of the Interactions Lab for their ideas. Thanks also to reviewers of an earlier version of this paper that was published at a conference. This work was funded by the National Science and Engineering Research Council of Canada, and by the Informatics Circle of Research Excellence.

REFERENCES

BAKER, K., GREENBERG, S. and GUTWIN, C. (2001): Heuristic evaluation of groupware based on the mechanics of collaboration. In LITTLE, M.R. and NIGAY, L. (Eds) Engineering for human-computer interaction (8th IFIP International Conference, EHCI 2001, Toronto, Canada, May), *Lecture Notes in Computer Science* 2254: 123–139, Springer-Verlag.

BAECKER, R. (1992): Readings in groupware and computer supported cooperative work. Morgan-Kaufmann, San Mateo, CA.

BAECKER, R., GRUDIN, J., BUXTON, W. and GREENBERG, S. (1995): Readings in human computer interaction: Toward the year 2000. Morgan-Kaufmann, San Mateo, CA.

BENFORD, S., GREENHALGH, C., BOWERS, J., SNOWDON, D. and FAHLÉN, L.E. (1995): User embodiment in collaborative virtual environments. In *Proceedings of CHI '95*, 242–249. Denver: ACM Press.

BIER, E. A. and FREEMAN, S. (1991): MMM: A user interface architecture for shared editors on a single screen. In *Proceedings of the 4th annual UIST,* 79–86. Hilton Head: ACM Press.
BOYLE, M. (2003): Collabrary shared dictionary v1.0.17: Programming paradigm and wire protocol. Report 2003-731-34, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada.
BOYLE, M. and GREENBERG, S. (2002): GroupLab collabrary: A toolkit for multimedia groupware. In PATTERSON, J. (Ed.). ACM CSCW 2002 Workshop on Network Services for Groupware, November.
BLY, S.A., HARRISON, S.R. and IRWIN, S. (1993): Media spaces: Bringing people together in a video, audio, and computing Environment. In *Communications of the ACM*, 3(1): 28–47.
CLARK, H. (1996): Using language. Cambridge University Press, Cambridge.
GAVER, W., MORAN, T., MacLEAN, A., LÖVSTRAND, L., DOURISH, P., CARTER, K. and BUXTON, W. A.S. (1992): Realizing a video environment: EuroPARC's RAVE system. In *Proceedings of CHI '92,* 27–34. Monteray, CA: ACM Press.
GOODWIN, C. (1981): *Conversational Organization: Interaction Between Speakers and Hearers*, Academic Press.
GREENBERG, S. and FITCHETT, C. (2001): Phidgets: Easy development of physical interfaces through physical widgets. In *Proceedings of UIST 2001,* 209–218. Orlando: ACM Press.
GREENBERG, S. and ROSEMAN, M. (2003): Using a room metaphor to ease transitions in groupware. In ACKERMAN, M., PIPEK, V., WULF, V. (Eds) Sharing Expertise: Beyond Knowledge Management 203–256, January, Cambridge, MA, MIT Press.
GREENBERG, S. and ROSEMAN, M. (1999): Groupware toolkits for synchronous work. In BEAUDOUIN-LAFON, M. (Ed.), *Computer-Supported Cooperative Work (Trends in Software 7)* 135–168. New York: Wiley.
GUTWIN, C. (1997): Workspace awareness in real-time distributed groupware. Ph.D Thesis, Dept of Computer Science, University of Calgary, Canada.
GUTWIN, C. and GREENBERG, S. (2002): A descriptive framework of workspace awareness for real-time groupware. Computer Supported Cooperative Work, 11(3-4): 411–446, Special Issue on Awareness in CSCW, Kluwer Academic Press.
HANSSON, P., WALLBERG, A. and SIMSARIAN, K. (1997): Techniques for "natural" interaction in multi-user CAVE-like environments. Poster in ECSCW '97.
ISHII, H. and KOBAYASHI, M. (1992): ClearBoard: A seamless medium for shared drawing and conversation with eye contact. In *Proceedings of CHI '92,* 525–532. Monterey: ACM Press.
JANCKE, G., GRUDIN, J. and GUPTA, A. (2000): Presenting to local and remote audiences: Design and use of the TELEP system. In *Proceedings of CHI 2000,* 384–391. The Hague: ACM Press.
KRUGER, R., CARPENDALE, M. S. T., SCOTT, S. D., and GREENBERG, S. (2003): How people use orientation on tables: Comprehension, coordination and communication. In *Proceedings of ACM Group 2000,* 369–378. Sanibel Island: ACM Press.
KRUGER, R., CARPENDALE, M.S.T., TANG, A., and SCOTT, S.D. (2004): Fluid orientation on a tabletop display: Integrating rotation and translation. Technical Report 2004-747-12, Department of Computer Science, University of Calgary, Calgary, Canada.
LOMBARD, M. and DITTON, T. (1997): At the heart of it all: The concept of presence. *Journal of Computer Mediated Communication*, 3(2), from http://www.ascusc.org/jcmc/vol3/issue2/lombard.html.
NARDI, B. A., WHITTAKER, S. and BRADNER, E. (2000): Interaction and outeraction: Instant messaging in action. In *Proceedings of CSCW 2000,* 79–88. Philadelphia: ACM Press.
RAMDUNY, D., DIX, A. and RODDEN, T. (1998): Exploring the design space for notification servers. In *Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW'98)* 227–235. Seattle: ACM Press.
ROGERS, Y. (1994): Exploring obstacles: Integrating CSCW in evolving organizations. In *Proceedings of CSCW '94,* 67–77. Chapel Hill: ACM Press.
ROSEMAN, M. and GREENBERG, S. (1996): Building real time groupware with GroupKit. *ACM Transactions on Computer Human Interaction*, 3(1): 66–106.
SHORT, J., WILLIAMS, E. and CHRISTIE, B. (1976): Communication modes and task performance. In *Readings in Groupware and Computer Supported Cooperative Work*, BAECKER, R. M. (Ed), Morgan-Kaufman Publishers, Moutain View, CA, 169–176.
STEWART, J., BEDERSON, B. B. and DRUIN, A. (1999): Single display groupware: A model for co-present collaboration. In *Proceedings of CHI '99,* 286–293. Pittsburgh: ACM Press.
TANG, J. (1991): Findings from observational studies of collaborative work. *Intl J Man-Machine Studies*, 34(2): 143–160.
TANG, J. and MINNEMAN, S. (1991a): VideoDraw: A video interface for collaborative drawing. *ACM TOIS Transactions on Information Systems*, 9 (2), April.
TANG, J. and MINNEMAN, S. (1991b): VideoWhiteboard: Video shadows to support remote collaboration. In *Proceedings of CHI '91,* 315–32. New Orleans: ACM Press.
TSE, E. and GREENBERG, S. (2002): SDGToolkit: A toolkit for rapidly prototyping single display groupware. In *Extended Abstracts of CSCW 2002,* 173–174. New Orleans: ACM Press.

BIOGRAPHICAL NOTES

Anthony Tang is an MSc student under the supervision of Dr Saul Greenberg in the Interactions Lab at the University of Calgary. His current research in Computer Supported Cooperative Work (CSCW) involves the study and development of tools for collaborative practices of co-located and distributed groups. More broadly, Anthony is interested in novel interaction techniques, design methodologies, and information visualisation.

Anthony Tang

Michael Boyle is a PhD student with Dr Saul Greenberg at the Interactions Lab in the Department of Computer Science at the University of Calgary. In his research, Mike seeks to inform the design of privacy-preserving video media spaces: always-on audio/video networks for casual interactions and informal awareness. For this, he has assembled a comprehensive descriptive theory of privacy and technology design, drawing on literature in environmental psychology, law, and computer science. Mike produced the Collabrary toolkit for rapidly prototyping distributed multimedia groupware applications. He has also interned at Microsoft Research, where he worked on prototype awareness and collaboration applications.

Michael Boyle

Saul Greenberg, a professor in Computer Science at the University of Calgary, is an active researcher in Human Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW). He and his group investigate how people work together, how the computer and related technologies (groupware) affect group behaviour, and how software and physical devices can be designed to support and augment group work. He and his team developed the now-commercialized Phidgets (Physical Widgets) that make it easy for the hardware-naive to rapidly prototype physical user interfaces. Similarly, he and his team developed many toolkits that make it easy for people to rapidly prototype multimedia groupware and single display groupware.

Saul Greenberg