From Awareness to TeamRooms, GroupWeb and TurboTurtle: Eight Snapshots of Recent Work in the GroupLab Project

Andy Cockburn
Department of Computer Science
University of
Christchurch, New Zealand
Tel: +64-3-364-2774
E-Mail: andy@cosc.canterbury.ac.nz

Saul Greenberg, Carl Gutwin, Mark Roseman
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
Tel: +1-403-220-6087
E-Mail: [saul, gutwin, roseman] @cpsc.ucalgary.ca

Cite as either:

- individual paper citation published in ACM SIGCHI 96 Companion Proceedings (details are provided at the top of each paper's page)
  or
- Greenberg, S., Gutwin, C., Roseman, R. and Cockburn, A. (1995) From Awareness to TeamRooms, GroupWeb and TurboTurtle: Eight Snapshots of Recent Work in the GroupLab Project. Research Report 95/580/32, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4, December.

Abstract

This report contains eight short papers that serve as snapshots of recent work by members and collaborators of the GroupLab team. All papers are concerned with groupware, and all but one of the systems described were implemented using GroupKit, our groupware toolkit.
Note: in this HTML version, the short papers are available independently by chasing the links below.

The first five papers are a suite of articles that considers how awareness of others can be supported in groupware systems. The papers cover theoretical considerations of awareness (#2), practical efforts in building systems and widgets to support awareness (#1,3,4) and evaluation of widgets to determine their effectiveness and usability (#5).

The next three papers cover individual projects. TeamRooms (#6) is a groupware equivalent of a physical team room. Group members can stock the room with applications, and can enter the room at any time to continue their work individually or collectively. GroupWeb (#7) is a a World Wide Web browser that is group-aware. People can share their views of pages in real time, can gesture around it with telepointers, and can add group annotations to a page with a groupware editor. TurboTurtle (#8) is a microworld for Newtonian physics designed for children. Children were observed using TurboTurtle, and their collaboration styles are analyzed.

Supporting Awareness of Others in Groupware

Carl Gutwin, Saul Greenberg, and Mark Roseman
Department of Computer Science, University of Calgary
Calgary Canada T2N 1N4
Tel: +1 403 220-6015
E-mail: [gutwin, saul, roseman]@cpsc.ucalgary.ca

INTRODUCTION

Successful collaboration in a work community requires that members of the group maintain awareness of one another on several levels. The following papers consider kinds of awareness and how these can be supported in groupware systems. In particular, the papers explore two kinds of awareness: first, general awareness of people in a work community, and second, awareness of others' interactions with a shared workspace.

In everyday work, *informal awareness* involves knowing who's currently around, whether they're available or busy, and what sort of activity they're engaged in. People need informal awareness in order to find opportunities for collaboration and people to collaborate with [1]. Once they have begun a collaborative activity using a shared workspace (such as a whiteboard or a document), they then need to maintain *workspace awareness*: where in the space others are working, what they are doing, and what changes they are making [2].

These same needs also exist when people work together through groupware. However, the cues and mechanisms that help people maintain awareness of others in face-to-face activity are often difficult to provide in groupware systems. For example, a virtual community becomes invisible on the network, and it is hard to see who is available for interaction. Also, glancing over at another part of a workspace to monitor another person's activities does not translate well to a groupware environment, especially if relaxed-WYSIWIS view sharing is in effect.

CSCW research has considered some of the issues surrounding awareness in cooperative work: general awareness of a work community has been supported through media spaces, and certain elements of awareness within a workspace have been supported with specialized displays in groupware systems. The papers in this suite build on these results and explore various facets of how awareness of others can be supported in groupware systems. Together, the papers cover theoretical considerations, practical efforts in building systems and widgets to support awareness, and evaluation of widgets to determine their effectiveness and usability.

CHI '96 Companion, Vancouver, BC Canada
© 1996 ACM 0-89791-832-0/96/04..$3.50

The first paper concerns informal awareness, and describes a system that provides information about people's whereabouts and activities, without requiring video connections. The second paper looks at awareness requirements of collaborative interaction in a shared workspace. It describes workspace awareness and organizes several elements into a conceptual framework that can be used as the basis for building groupware support.

The next two papers describe inventions that have been built as experimental supports for workspace awareness. The first of these presents a class of widgets called radar views, and the second discusses how fisheye visualization techniques can be used as a basis for providing awareness information in groupware.

The final paper reports on a study carried out to evaluate the usability and effectiveness of several awareness widgets in a realistic groupware application. The study reinforces the idea that workspace awareness is maintained and used in collaborative activity, and found that some widget designs were successful in providing some of the awareness information that participants needed.

The experiences gathered within these papers argue that awareness issues must be considered more generally in the design of groupware systems. In addition, further exploration of how awareness works can help groupware systems better support the natural, facile interactions that mark everyday face-to-face collaboration.

More information about the research discussed in these papers can be found at the GroupLab web site: http://www.cpsc.ucalgary.ca/projects/grouplab/

REFERENCES

1. Kraut, R., Egido, C., and Galegher, J. Patterns of Contact and Communication in Scientific Research Collaborations. In *Intellectual Teamwork*. J. Galegher, R. Kraut, and C. Egido eds. LEA Press, 1990, 149-172.
2. Gutwin, C., Stark, G., and Greenberg, S. Supporting Workspace Awareness in Educational Groupware. In *Proc. CSCL'95*, Bloomington, IN, (1995), pp. 147-156.

Peepholes: Low Cost Awareness of One's Community

Saul Greenberg
Department of Computer Science, University of Calgary
Calgary, Alberta CANADA T2N 1N4
Tel: +1-403-220-6087
E-mail: saul@cpsc.ucalgary.ca

ABSTRACT
In distributed communities, media spaces supply people with an awareness of who is around by displaying video or periodic snapshots of common areas and offices. This in turn facilitates casual interaction. Peepholes is a low cost alternative. Instead of video, iconic presence indicators show the availability of people in a virtual community. If people are absent, a user can 'ambush' them by asking the system to announce their return. When interaction is desired, people can easily contact one another because communication software is just a button-press away.

Keywords
Groupware, contact facilitation, awareness, coordination.

INTRODUCTION

Informal awareness of one's community is the general sense of who is around and what others are up to—the kinds of things that people track when they work together in the same physical environment. This awareness is the glue that facilitates casual interaction, the spontaneous and one-person initiated meetings that form the backbone of everyday coordination and work [1,3]. Yet casual interaction is problematic in distributed communities. While groupware is readily available, people have considerable trouble staying aware of opportunities for collaboration, and in establishing electronic meetings.

Media spaces are one way of providing distributed groups with informal awareness of each other. Users can select offices and common areas at remote sites, and view them through continuous video. Yet even compressed video demands too much bandwidth for everyday use. Portholes [2] partially solves this problem by periodically transmitting small video snapshots instead of a video stream. The community is presented on one's screen as an array of images. However, Portholes still requires people to have video cameras attached to their workstations and a willingness to leave them turned on.

An alternative to video is iconic *presence indicators* that show who is around and the likelihood of their availability. This paper shows how presence indicators, as implemented in Peepholes, afford casual interaction, especially when they are integrated with common communication and groupware facilities.

PEEPHOLES

Creating a virtual work community.

Users can create their virtual community in Peepholes by choosing potential collaborators from an electronic address book. For example, Figure 1 shows a virtual community of five people selected from the book. Each person is represented by a labeled icon and optional address entry.

Peepholes automatically maintains the address book. It first scans incoming email for names and electronic addresses, and then adds new members to the book or updates old ones. This tracks people who have recently communicated with the user, and who are likely to be part of one's virtual community. In the book, a user can create sub-communities and assign people to them. Particular sub-communities can then be recalled quickly.

Informal awareness through activity indicators.

Opportunities for casual interaction happen when people are aware that others are available for communication. On a network, similar opportunities could occur if we could see who is actively working at their computer. Because computers can easily capture and transmit how long it has been since their users were active, this information can be displayed as an estimate of a person's real availability. For example, the Peephole icons in Figure 1 continually display the activity status of each community member. Greenberg is now active (denoted by a bold character), O'Grady has been idle for a few minutes (the grayed out icon), Lowe is logged on but hasn't used the computer in a while, Schaffer is logged off, and Roseman is unreachable. A quick glance at these icons gives awareness of people's probable availability for real time communication.

Ambushes for tracking availability.

It is not easy for one person to initiate a meeting over distance, as people are often absent or not immediately available. While activity indicators suggest when a call will work, they must be monitored regularly to see when an absent person to return. Indeed, users of the Cruiser media space would often open a full bandwidth video connection

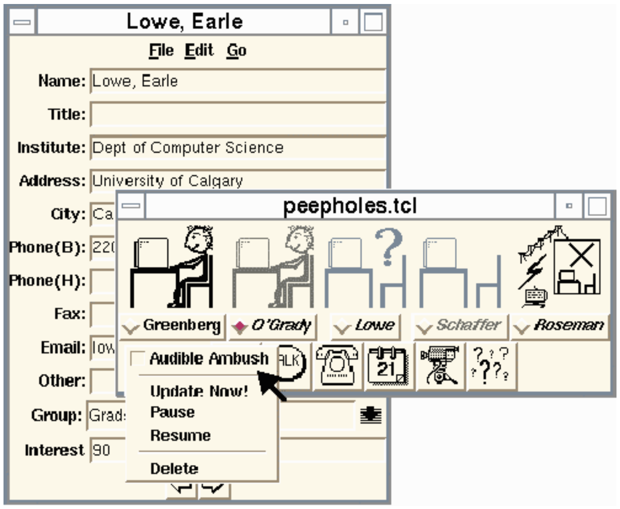

Figure 1. Peepholes activity icons and an address entry

to the empty office of a collaborator, solely to 'ambush' its occupant, i.e., to see when they returned [3]. In Peepholes, users can ambush others through a menu option (Figure 1). When the system notices that the person has become active, it announces their return by playing an audible sound of someone typing. This attracts the ambusher's attention to the display, allowing them to initiate a call if desired.

From informal awareness to making contact.
Moving from awareness of another's availability to an informal meeting is simple in physical environments but not on computers. Electronic addresses must be found, software connections established, system compatibility verified. Peepholes simplifies this by integrating communication and groupware tools via hooks. Electronic addresses are maintained in the address book, and connections established by simply selecting a person's Peephole icon and an application icon (Figure 1, bottom). Connections are literally a button click away.

From asynchronous to casual interaction.
Email represents an opportunity for casual real-time interaction that should not be ignored. Peepholes is linked to a mail reader and, as people read their mail, a Peephole icon is automatically raised on the sender (Figure 2). The user can see if the sender is available for real time conversation, and contact them that way if desired. This is particularly useful for incoming mail, where it is more than likely that the sender is still active on their machine.

Information for free.
Peepholes only uses information freely available on the network [1]. No specialized software is installed at remote sites, allowing it to be used anywhere on the Internet. It works by continually querying the *ruser daemons* found on most Unix-based servers, and by massaging the results. In practice, this is a reasonable way to access many users.

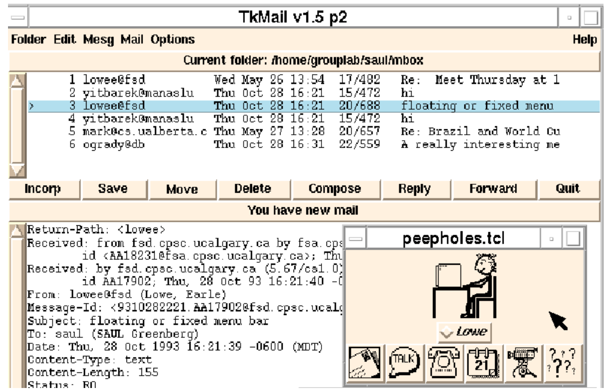

Figure 2. Integrating Peepholes and electronic mail

DISCUSSION AND SUMMARY

In actual use, Peepholes does let a user maintain informal awareness and establish contact with others. As only a few bits of information are transmitted and no special equipment required (*cf. video*), it is very low cost. Although activity indicators cannot tell the difference between absent and inactive people, they are reasonable indicators of another's availability. The ambush feature is a surprisingly effective way of getting hold of another person. Establishing connections is straight-forward, although software incompatibilities do occur. Information "for free" is useful but limited: some people are not observable because some sites do not install the ruser daemon, or use restricted versions of them, or insulate themselves from the outside world through firewalls.

We are now taking the ideas in Peepholes and installing them as components used by session managers in GroupKit, our groupware toolkit. Peephole awareness will be the lowest common denominator used to facilitate casual interaction. However, the system will also check for more sophisticated capabilities and substitute them when appropriate. For example, Peephole icons could be progressively replaced by participants' images, by periodic snapshots, or even by full video. If custom daemons are used, they can better track awareness information, and can allow people to control the degree of privacy desired.

REFERENCES

1. Cockburn, A. and Greenberg, S. (1993). Making contact: Getting the group communicating with groupware. *Proc ACM Conference on Organizational Computing Systems*, Nov. 1-4.

2. Dourish P. and Bly S. (1992). Portholes: Supporting awareness in a distributed work group. *Proc ACM Conference on Human Factors in Computing Systems*, pp. 541-547, May 3-7.

3. Fish R., Kraut R., Root R. and Rice R. (1992) Evaluating video as a technology for informal communication. *Proc ACM Conference on Human Factors in Computing Systems*, pp. 37-48, May 3-7.

Workspace Awareness for Groupware

Carl Gutwin and Saul Greenberg
Department of Computer Science, University of Calgary
Calgary Canada T2N 1N4
Tel: +1 403 220-6015
E-mail: [gutwin, saul]@cpsc.ucalgary.ca

ABSTRACT
Shared physical workspaces allow people to maintain up-to-the minute knowledge about others' interaction with the workspace. This knowledge is *workspace awareness*, part of the glue that allows groups to collaborate effectively. In this paper, we present the concept of workspace awareness as a key for groupware systems that wish to support the fluid interaction evident in face-to-face collaboration. We discuss why workspace awareness is difficult to support in groupware systems, and then present a conceptual framework that groupware designers can use as a starting point for thinking about and supporting awareness.

KEYWORDS: Workspace awareness, groupware, CSCW

INTRODUCTION

Shared physical workspaces (such as a chalkboard, a control panel, or a tabletop) and the artifacts in them act as stage and props for rich person-to-person interaction (e.g. [1]). The affordances of physical workspaces allow people to maintain awareness of others' locations, activities, and intentions relative to the task and to the space—awareness that enables them to work together more effectively. We call this *workspace awareness*: the collection of up-to-the minute knowledge a person uses to capture another's interaction with the workspace.

Real-time distributed groupware often provides shared *virtual* workspaces. However, interactions within virtual workspaces are impoverished when compared with their physical counterparts. We want to enrich this interaction, and so we are exploring the concept of workspace awareness. The following sections describe workspace awareness, outline the problems faced in supporting it, and present a framework that organizes knowledge about the concept into a form usable by groupware designers.

WORKSPACE AWARENESS

In our own observational studies of collaboration over physical workspaces, we have looked at how workspace awareness operates in mixed-focus situations, where group members shift their attention back and forth between individual and shared activity. In these situations, the workspace allows lightweight information gathering such as quick glances over at another person's work area. This information is integrated with existing knowledge to maintain a sense of awareness of where the other person is and what they are doing. Workspace awareness aids coordination of tasks and resources, and assists transitions between individual and shared activities. People can use their knowledge to anticipate others' actions, assist them with their tasks, and interpret deictic references to objects. The benefits of workspace awareness are subtle, but over the course of a collaborative interaction, they can markedly improve a group's effectiveness.

The Problem of Workspace Awareness in Groupware

Workspace awareness comes naturally in a face-to-face situation, but it is far more difficult to maintain in a real-time groupware system. In groupware, people may only see a fraction of the workspace, and may not see the same part as other group members. A groupware system also reduces the richness of communication, and its interface may hide many actions that are visible in a physical workspace. Furthermore, perceptual and physical abilities that we use to maintain workspace awareness (such as glances) are often replaced with mechanisms that are comparatively slow and clumsy (such as scrolling).

Within this different environment, the groupware designer must try and recreate the conditions and cues that allow people to keep up a sense of workspace awareness. Whereas face-to-face interaction has inherent mechanisms and affordances for maintaining workspace awareness, the groupware designer is faced with a blank slate—any support for building or maintaining workspace awareness must be explicitly determined and built into the groupware system, and it is not obvious what that support should be.

A FRAMEWORK OF WORKSPACE AWARENESS

Groupware designers face two problems in designing awareness support. First, what information should a groupware system capture about another's interaction with the workspace? Second, how should this information be presented to other participants? We have built a framework of workspace awareness to address these issues. It presents a set of basic ideas that are central for designing awareness

CHI '96 Companion, Vancouver, BC Canada
© 1996 ACM 0-89791-832-0/96/04..$3.50

support, and that allow different techniques to be identified, described, and compared. The framework considers both the elements that make up people's workspace awareness, and the mechanisms they use to gather awareness information.

Element	Relevant Questions
Presence	Who is participating in the activity?
Location	Where are they working?
Activity Level	How active are they in the workspace?
Actions	What are they doing? What are their current activities and tasks?
Intentions	What will they do next? Where will they be?
Changes	What changes are they making, and where?
Objects	What objects are they using?
Extents	What can they see? How far can they reach?
Abilities	What can they do?
Sphere of Influence	Where can they make changes?
Expectations	What do they need me to do next?

Table 1. Elements of workspace awareness

The table shows a set of elements that we consider to be part of workspace awareness, and lists questions that a participant might ask themselves during a shared activity. Many of the elements fall into two rough groups: those that deal with *what* is happening with another person (e.g. amount of activity, nature of actions, changes, and expectations), and those that deal with *where* it is happening (location of focus, view extents, area of influence, or objects in use).

These elements provide a basic vocabulary for thinking about awareness requirements and groupware support. Designers can use the framework to analyze existing face-to-face situations. As a simple example, a group activity like a jigsaw puzzle may require that people stay aware of where in the puzzle others are working, but not the particular objects that they are manipulating. In addition to considering which elements are more or less important in a particular situation, there are several ways that a designer can assess how elements are used. For example:

- elements may consider a person's interaction with the workspace in the past instead of the present (e.g., where others have been and what they have been doing);
- elements may constrain one another (e.g., someone's location may also indicate what they are doing)
- elements may imply different information granularity (e.g. in loosely-coupled collaboration, people may need only a general idea of where others are working).

The framework also considers how people gather information to maintain workspace awareness. However, determining precise mechanisms in face-to-face situations is difficult, since they can be subtle, hard to observe (sound cues, for example), or buried within layers of inference. Instead, we have determined a general set of information-gathering mechanisms that have been discussed in previous literature, basic mechanisms through which workspace awareness is maintained.

- *Direct communication*: explicit communication through speech or gesture [1], often employing deictic reference.
- *Indirect productions:* utterances, expressions, or actions that are not explicitly directed at others, but that are intentionally public.
- *Consequential communication*: the visible or audible signs of interaction with a workspace [4]. Watching someone work provides clues about their actions.
- *Feedthrough*: the observable effects of someone's actions on the workspace's artifacts. Seeing an object move indicates that someone is moving it.
- *Environmental feedback:* feedback from the environment or overall workspace caused by the indirect effects of someone's actions.

Groupware designers must consider how information about various elements is transmitted and gathered, and must allow people to continue using natural mechanisms like those listed above, or others specific to particular domains and situations. With knowledge of these mechanisms, and of how they are used to maintain different elements of awareness, a designer can begin to create techniques and widgets that provide people with appropriate information about others in a virtual workspace.

CONCLUSION

Workspace awareness is an important concept for real-time distributed groupware. By setting out elements and mechanisms of workspace awareness, the conceptual framework above provides a vocabulary and a starting point for thinking about and designing groupware support. We currently use the framework to inform the design of awareness widgets for a groupware toolkit. In future, we plan to expand and validate the framework through additional studies of face to face groups, to continue building awareness widgets for particular situations, and to investigate other issues raised in applying the framework to groupware.

REFERENCES

1. Tang, J. Findings from Observational Studies of Collaborative Work. *IJMMS* 34,2 (1991), 143-160.
2. Gutwin, C., Stark, G., and Greenberg, S. Supporting Workspace Awareness in Educational Groupware. In *Proc. CSCL'95*, Bloomington, IN, (1995), 147-156.
3. Dourish, P. and Bellotti, V. Awareness and Coordination in Shared Workspaces. In *Proc. CSCW'92* (Toronto, 1992), 107-114.
4. Segal, L. Actions Speak Louder Than Words: How Pilots Use Nonverbal Information. In *Proc. Human Factors '94* (1994), 21-25.

Workspace Awareness Support With Radar Views

Carl Gutwin, Saul Greenberg, and Mark Roseman
Department of Computer Science, University of Calgary
Calgary Canada T2N 1N4
Tel: +1 403 220-6015
E-mail: [gutwin, saul, roseman]@cpsc.ucalgary.ca

ABSTRACT

Real-time groupware systems often let each participant control their own view into a shared workspace. This strategy can reduce awareness about where and how others are interacting with the document or the workspace artifacts. We have designed a number of awareness widgets to help people regain this awareness. In this paper we present several radar views that provide awareness information on top of a global overview of the workspace. Our displays give lightweight access to information about others' locations and activities, providing for richer person-to-person interaction in groupware systems.

KEYWORDS: Radar views, widgets, workspace awareness

INTRODUCTION

For several years, we have been building real-time groupware systems with the GroupKit toolkit. These systems are often based on shared workspaces like whiteboards, written documents, or design drawings. As our work evolved, we recognized the importance of workspace awareness as a quality that can improve groupware systems.

Workspace awareness is the up-to-the minute understanding of how another person is interacting with a workspace, and includes knowledge about where people are working, what they are doing, the changes they are making, and their future intentions [1]. By supporting workspace awareness, groupware systems can better allow the rich interaction evident in face-to-face collaboration over physical workspaces.

However, supporting workspace awareness in groupware can be difficult, especially when participants are allowed to work in different parts of the workspace (relaxed-WYSIWIS view sharing) [2]. In these situations, people cannot directly see what others are doing, and may lose track of where they are in the space. To support awareness in our relaxed-WYSIWIS systems, we have built a variety of widgets that provide people with information about others' interaction with the groupware workspace.

Our goal in designing these widgets has been to provide information that allows people to collaborate better, but without hampering each person's local activity. In addition, limitations on screen space require that our designs not consume a large area.

RADAR VIEWS FOR WORKSPACE AWARENESS

Radar views are a class of widgets that are based on miniature overviews of an entire workspace. These miniatures have been seen in video games and some groupware systems (e.g. [3]). Because the overviews show the entire workspace, they are a natural vehicle for awareness information in relaxed-WYSIWIS situations. We have constructed four widgets based on the miniature overview. These inventions are the radar view, the history radar, the portrait radar, and the head-up radar. Each widget is described below.

Radar views

Our "radar" view (see Figure 1) adds information about other people's interaction to a basic overview. Since the overview already provides a spatial representation of the workspace, we have first added information about where others are working. The radar display shows what each person can see, by marking view outlines (first seen in

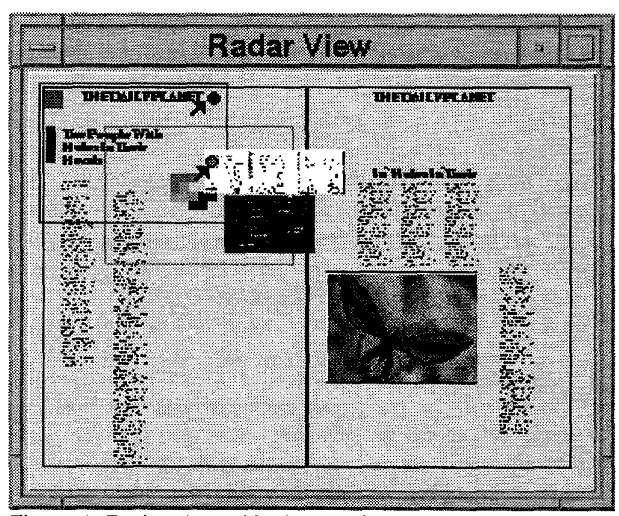

Figure 1. Radar view with view outlines and telepointers

CHI '96 Companion, Vancouver, BC Canada
© 1996 ACM 0-89791-832-0/96/04..$3.50

SASSE's text overview [3]), and also shows finer-grained location by including miniature telepointers that represent each person's mouse cursor. These additions support awareness of another person's general and specific location in the workspace.

The radar view also supports awareness of activity. Since the radar shows movement of and changes to artifacts in the workspace, it already provides some information about others' actions. Adding telepointers to the display provides a second source of information about what people are doing. In addition, it is easy for groupware designers to provide task-specific feedback about types of activity, such as selection of objects or use of different tools.

Portrait radar

The basic radar view shows each participant's view outline and telepointer in a unique colour. One problem with this approach is that it can be difficult to sort out which view rectangle belongs to whom. To simplify interpretation, we attach names or portraits to the view rectangles, which allows more natural identification (see Figure 2). In future, we plan to replace these static pictures with video images.

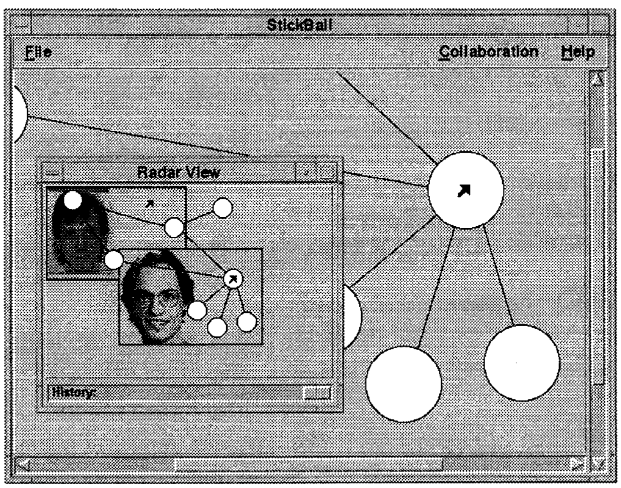

Figure 2. Portrait radar with history. Moving the radar's slider shows where viewports have been in the past.

The portraits sit behind the artifacts in the display, so this widget is most useful in sparse workspaces or where artifacts are transparent.

History Radar

We have also experimented with supporting awareness of people's past locations as well as their current position. Awareness of past location can be valuable, for example, if people need to determine the parts of a workspace that others have already visited. We record participants' view positions as a session progresses, and have added a slider to the radar view that allows people to "roll back" time and see where others have been. In Figure 2, the slider is shown at the bottom of the radar window. Dragging it to the left replays past locations of the view outlines.

Head-up radar

In some cases, screen space may be at such a premium that there is no room for a separate radar view. To minimize space, we are experimenting with a "head-up" display that combines normal and radar views. The widget in Figure 3 shows the full-size viewport as the front layer, and a miniature of the entire workspace as the back layer, coloured grey to reduce distraction. The rectangles in the background show the extents of each person's detail view.

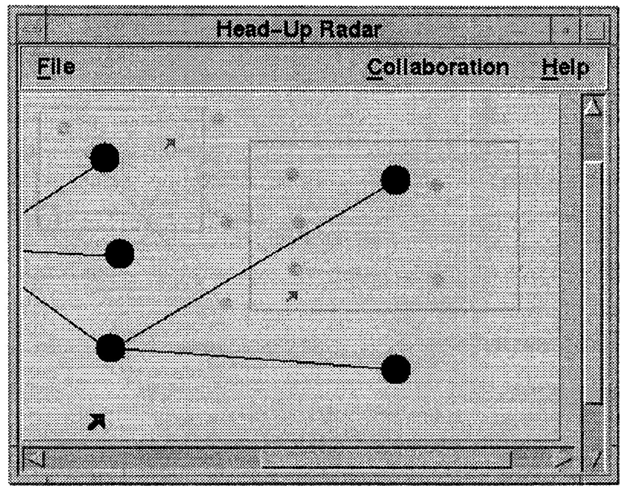

Figure 3. Head-up radar view. The detail view (in black) is overlaid on the radar view (in grey).

The head-up radar view is able to provide workspace awareness information without requiring the user to look at a separate display. One question we plan to explore through user testing is whether people can easily separate the two layers of the display. Again, this widget will work best in workspaces where artifacts are sparse.

CONCLUSION

We have begun evaluating the usability and effectiveness of these widgets, by testing them in realistic groupware applications. In addition, we are working on new designs that support other aspects of workspace awareness in relaxed-WYSIWIS groupware.

REFERENCES

1. Gutwin, C., Stark, G., and Greenberg, S. Supporting Workspace Awareness in Educational Groupware. In *Proc. CSCL'95*, Bloomington, IN, (1995), pp. 147-156.
2. Stefik, M., Bobrow, D., Foster, G., Lanning, S. and Tatar, D. WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *Trans. Office Information Systems* 5, 2 (1987), 147-167.
3. Baecker, R., Nastos, D., Posner, I. and Mawby, K. The User-Centred Iterative Design of Collaborative Writing Software. In *Proc. INTERCHI'93* (1993), pp. 399-405.

A Fisheye Text Editor for Relaxed-WYSIWIS Groupware

Saul Greenberg
Department of Computer Science, University of Calgary
Calgary, Alberta CANADA T2N 1N4
Tel: +1-403-220-6087
E-mail: saul@cpsc.ucalgary.ca

ABSTRACT
Participants in a real-time groupware conference require a sense of awareness about other people's interactions within a large shared workspace. Fisheye views can afford this awareness by assigning a focal point to each participant. The fisheye effect around these multiple focal points provides peripheral awareness by showing people's location in the global context, and by magnifying the area around their work to highlight interaction details. An adjustable magnification function lets people customize the awareness information to fit their collaboration needs. A fisheye text editor illustrates how this can be accomplished.

Keywords
Groupware, fisheye views, awareness, visualization.

INTRODUCTION

Real-time distributed groupware typically provides a shared virtual workspace where people can see and manipulate work artifacts. Many systems now follow a relaxed "what-you-see-is-what-I-see" (*relaxed-WYSIWIS*) model, where people can have different viewports into the workspace. The problem is that groupware workspaces do not yet afford the richness of interaction available in their physical counterparts. In particular, it is more difficult to maintain a sense of *workspace awareness*: the up-to-the-minute knowledge about another person's interactions with the shared workspace [1]. In groupware, people's normal mechanisms for tracking what goes on around them, such as peripheral vision and quick glances, are ineffective since the required information may be absent from the display. People can lose the sense of awareness that is essential for coordinating interaction, such as where others are operating and what they are doing.

One solution supplies users with two separate windows: a full sized viewport, and a *radar overview* that presents an active miniature of the workspace. Radar views typically overlay boxes atop the miniature to indicate other participant's viewport [1]. However, radar views introduce a seam between local and global contexts. To gather awareness information, people must attend to and mentally integrate two displays that differ in both scale and physical location. As well, the actions shown in the miniature may not be useful due to the loss of resolution and detail.

Applying fisheye visualization techniques to groupware can remove this seam. In conventional fisheye systems, multiple focal points magnify regions of personal interest within the global context [3]. A groupware fisheye twists this notion by assigning a focal point to each participant. Consequently, a person's view into the shared workspace will contain magnified regions showing others' work areas, seamlessly integrated into the global context. These regions provide awareness of the details of others' actions. If the magnification function is adjustable, a person can even customize the awareness information to suit their particular collaboration needs in the shared workspace.

A FISHEYE TEXT EDITOR USED BY ONE PERSON

A fisheye view is a visualization technique that provides both local detail and global context in a single display. It takes its name from camera lenses that distort a scene to provide very wide angles of view. In a computational fisheye, the user chooses one or more points of focus where they wish to see local detail [3]. These areas are visually emphasized, with the surrounding regions de-emphasized by graphical scaling, filtering, or clustering.

The editor in Figure 1 uses a fisheye lens to present a text document. Most of the text is shown at a very small font, which gives the person a sense of the document's global structure. The user views local detail by selecting a focal point in the document with the mouse or scrollbar. Fisheye effects are customized through a novel *lens widget*—users can resize the background text, and modify the shape of the lens that magnifies text around their focal point.

A GROUPWARE FISHEYE TEXT EDITOR

While valuable for single-user information visualization, fisheye views can be extended to support workspace awareness in groupware as well. The fisheye text editor above is actually a groupware application constructed in a groupware toolkit [2]. As groupware, the editor lets multiple people view the same document. People join into it through a session manager (Figure 2, top right). Although

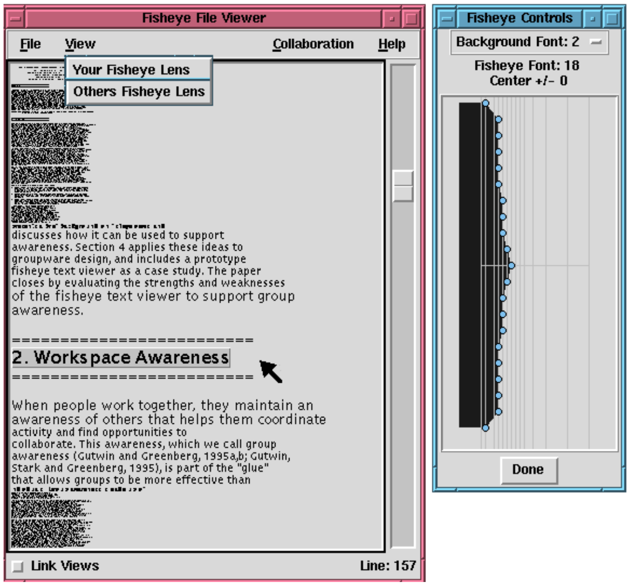

Figure 1. The fisheye text viewer with one user

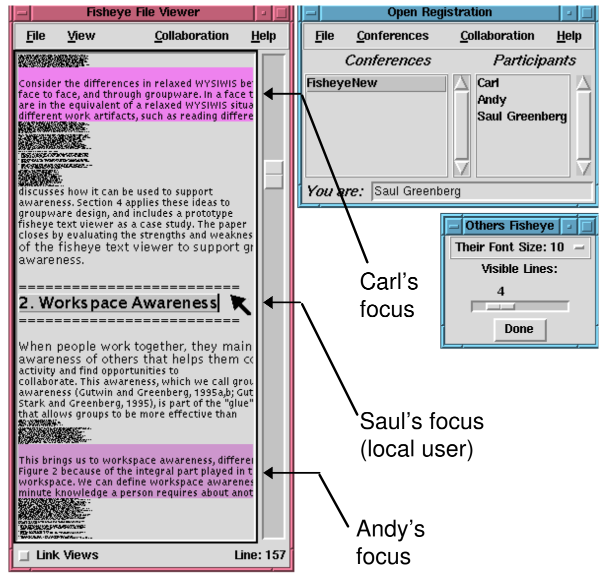

Figure 2. The viewer showing three participants.

the same document is visible on all displays, views are relaxed-WYSIWIS: each person can set their own focal point and customize the fisheye effect.

Support for workspace awareness involves representing each participant's focus in the document. First, location information and user identification is presented by marking others' focal points with an assigned color. Second, the text around other participants' focal points is also magnified. Figure 2 shows three focal points with corresponding magnified regions; the center region belongs to the user and the surrounding two represent the other participants. Their locations in the global context and the details of their work are clearly visible.

As people move between loosely and tightly coupled collaborations, their awareness requirements will change. Because display space is at a premium, a person should be able to allocate screen space for their own work or for the display of awareness information, as required. In the fisheye editor, a person can customize the amount of awareness detail by altering the magnification function applied to others' focal points, by changing the background magnification, and by linking their views.

- If only location information is desired, one can turn off the magnification of other participants' focal points. Others' locations remain visible through color, but no extra screen space is used.
- For finer-grained awareness, the detail visible can be progressively increased by growing the magnification around the other participant's focus, as well as the extent of the region being magnified.
- When people are working far apart in the document, a "split window" effect can bring them closer together. The background font is made invisible, thus displaying only the regions surrounding each focal point.
- For tightly-coupled collaboration, people can link their views, which lets all participants share a common focal point. If any user changes the focus, it will be changed on all other displays as well.

DISCUSSION AND SUMMARY

There is more to awareness than knowing other's location and actions, and an appraisal of the fisheye editor against an awareness framework [1] has identified other requirement that need to be addressed. For example, because people need to be aware of others' movements and gestures in the workspace, we have added telepointers to the system. We also need a better way of identifying who belongs to a given focal point, as color is not a particularly good cue. However, the basic fisheye concept seems sound.

We believe that fisheye views are a good approach to help people maintain workspace awareness. This remains to be confirmed by user-based studies.

REFERENCES

1. Gutwin, C., Stark, G., and Greenberg, S. (1995) Support for group awareness in educational groupware. In *Proc Computer Supported Collaborative Learning*, pp. 147-156, Bloomington, IN, Oct 17-20, LEA Press.
2. Roseman, M. and Greenberg, S. (1996) Building real time groupware with GroupKit, a groupware toolkit. *ACM Trans Computer Human Interaction*, in press.
3. Sarkar M., Snibbe S., Tversky O., and Reiss S. (1993) Stretching the rubber sheet: A metaphor for visualizing large layouts on small screens. In *Proc ACM User Interface Software and Technology*, pp. 81-91.

A Usability Study of Workspace Awareness Widgets

Carl Gutwin and Mark Roseman

Dept. of Computer Science, University of Calgary
Calgary, Alta, Canada T2N 1N4
Tel: +1-403-220-3532
E-mail: gutwin, roseman@cpsc.ucalgary.ca

ABSTRACT

Groupware systems that use large shared workspaces generally provide only limited awareness information about other collaborators in the workspace. We are designing a set of groupware widgets to provide this missing information. This paper describes a usability study of a number of such widgets. The study has both validated our intuitions about the need for workspace awareness information, and revealed the strengths and weaknesses of several current designs.

KEYWORDS: Groupware, CSCW, awareness, usability

INTRODUCTION

Compared with physical shared workspaces such as table tops and whiteboards, shared workspaces in groupware are greatly impoverished. In particular, systems supporting a relaxed-WYSIWIS (What You See Is What I See) view of large workspaces often fail to convey information about *workspace awareness*, the up-to-the minute knowledge about the location and actions of other collaborators.

We have designed a suite of *groupware awareness widgets* to address this deficiency [2]. These widgets augment a user's view of the workspace with information about the workspace, the location of other collaborators, and their actions within the workspace.

This paper describes initial observations from a study carried out to evaluate a shared workspace system that incorporated several different awareness widgets. We had two goals in this study. First, we wanted to confirm our intuitions that workspace awareness is used in shared workspaces. Second, we wanted to evaluate how well our current widget designs support the maintenance of workspace awareness. We were particularly interested in knowing if the information in the widgets was easy to interpret, if they distracted users from their tasks, and if users thought they were worth the extra screen space.

METHODOLOGY

We constructed a groupware editor for manipulating the layout of a two page newspaper spread, allowing users to move pictures, headings, and columns of text. Eight pairs of subjects, primarily senior undergraduate computer science students, worked on separate workstations. Subjects were within speaking distance but unable to see each other. Each user could scroll independently within the layout, and their window was large enough to view about one third of the workspace at a time.

Pairs completed two layout tasks, each limited to fifteen minutes. One task was performed with limited awareness information about the other person. Subjects used either the shared workspace by itself, or combined with a *miniature view* showing only the locations of objects within the workspace. Half the pairs completed this condition first.

In the other condition, subjects used the shared workspace along with one of three awareness widgets we had built. The first was a *multi-user scrollbar*, which shows the location of each user with a colored bar beside the "thumb" of the conventional scrollbar. The second was a *radar view*, which shows a miniature of the entire document, a rectangle for the extent of each user's view, and a telepointer showing their mouse location. In both of these widgets, participants are differentiated by color. The third was a *local view* widget, which shows the full scale but limited region immediately around the other user's mouse cursor.

To collect data, we videotaped the tasks, asked subjects to fill out questionnaires, and conducted interviews to follow up particular aspects of the session.

RESULTS

All pairs completed their tasks and produced reasonable layouts, and made use of workspace awareness in doing so. Several of the widgets that we tested provided useful awareness information, and most subjects greatly favoured the conditions that included these widgets.

Use of Workspace Awareness

We observed a variety of working styles, ranging from "divide and conquer" to tightly coupled collaboration. Regardless of the style, there was evidence that the pairs maintained an awareness of each other's use of the workspace, and acted on that information to collaborate with their partner and complete their task.

Many of our observations of the use of awareness echo previous observational studies. For example, we noticed the frequent use of gestures [3]. For gestures to be interpreted, the receiver must see them, and therefore the sender must be aware of the receiver's view. Gestures were most often communicated through the main telepointers, but people sometimes gestured by moving objects in the workspace or by using the telepointer in the radar widget. In addition to gestures, we also noticed the regular use of deictic references (e.g. "move *this* object"). As with gestures, deixis depends on the hearer being able to see what the speaker is pointing at, or the hearer having a mental picture of their work area.

Use of Awareness Widgets

The widgets were well received by subjects, who used them, liked them, and often requested even more awareness information than what was available. Widgets were used in two ways. First, subjects used the widgets to keep track of the locations of objects, and as a high level overview of the entire layout. For example, many subjects used the radar to check if text columns fit on the page. Second, subjects used the widgets to keep track of their partner's location, activities, and progress on the task. For example, widgets assisted subjects in discussing placement of articles with their partner, who was working on the other page.

Feedback from subjects showed that the radar and miniature view widgets were most useful in their task. Although subjects could see some use for the local view and multi-user scrollbar widgets, these did not seem to support the task better than the plain workspace. Below, we consider these results in terms of ease of interpretation, distraction and perceived value of the widgets.

To explore how easily the information in the widgets was interpreted, we considered the difficulty of shifting contexts between the main view and the widgets, and the problem of mapping colors to users. The context shift to the radar view proved not to be a problem—subjects reported that it was easy to identify workspace objects in the radar view. Users found it more difficult to integrate the two different dimensions of the scrollbar than to interpret the view rectangles in the radar. Interpreting the local view was extremely problematic for all subjects who used it. Several remarked on its small size and erratic motion. The mapping between colors and people in the radar and the multi-user scrollbar proved difficult for some subjects. One subject said "I couldn't figure it out; I just watched for motion."

Distraction was an issue in some widgets but not in others. None of the subjects found that the radar view or miniature stole their attention. Only one of four subjects found the scrollbars distracting. However, all users of the local view found it very disruptive, due to its erratic motion.

Finally, we asked subjects about the value of the widgets in completing their task, and whether they were worth their screen space. All subjects using the radar and miniature said that they found them valuable and worth the screen space — and even complained when we took them away. Their assessment of the scrollbars and local view was less positive; only two scrollbar users found them valuable, while none of the local view users found it valuable.

DISCUSSION

There are a number of lessons that can be drawn from these results. First, overviews are useful both for managing one's own interaction with a large workspace, and for maintaining awareness of other's locations and activities. The overview frees the user from having to maintain a mental model of the ever-changing workspace found in relaxed-WYSIWIS groupware. Second, if awareness information is to be easily interpreted, it must be presented in a familiar context, ideally that of the workspace itself [1]. Though the radar was physically separate, it closely paralleled the workspace, providing an easy transition between the two. Third, awareness widgets must try to be as lightweight as the mechanisms of face-to-face interaction, or they will not be used. For example, integrating the two separate dimensions of the scrollbar required more effort than finding view outlines in the radar.

CONCLUSIONS

Overall, this study has confirmed our belief that workspace awareness is an important part of collaborating in large workspaces. Though all of our pairs completed the task, we found that some of our widget designs provided useful awareness information that would otherwise be missing from a groupware system. This information allowed for more natural interaction over the workspace. In some cases, our widgets were remarkably effective, leading one subject to remark "it really felt like you were working on the same big table." We expect the issues raised here will motivate groupware designers to continue exploring awareness with the goal of building more natural shared workspaces.

REFERENCES

1. Dourish, P., and Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proc. Computer Supported Cooperative Work (CSCW '92)*.
2. Gutwin, C., Stark, G., and Greenberg, S. (1995). Support for workspace awareness in educational groupware. In *Proc Computer Supported Collaborative Learning*, Bloomington, IN, Oct 17-20. LEA Press.
3. Tang, J. (1991). Findings from observational studies of collaborative work. *IJMMS*. **34**(2).

GroupWeb: A WWW Browser as Real Time Groupware

Saul Greenberg and Mark Roseman
Department of Computer Science, University of Calgary
Calgary, Alberta CANADA T2N 1N4
Tel: +1-403-220-6087
E-mail: {saul, roseman}@cpsc.ucalgary.ca

ABSTRACT
GroupWeb is a browser that allows group members to visually share and navigate World Wide Web pages in real time. Its groupware features include document and view slaving for synchronizing information sharing, telepointers for enacting gestures, and relaxed "what you see is what I see" views to handle display differences. A groupware text editor lets groups create and attach annotations to pages. An immediate application of GroupWeb is as a presentation tool for real time distance education and conferencing.

Keywords
Word wide web, groupware, shared visual workspaces.

INTRODUCTION

The World Wide Web is becoming a universal repository for distributed information, with Web browsers becoming the standard way that people search for and display items of interests. Although the information is a shared resource authored by the community, Web browsers are still single user tools that partition one person from another.

What if Web browsers were redesigned as groupware that allowed people to share views of pages? Such a browser would be a valuable presentation tool for real time distributed meetings for several reasons. First, relying on a user's established Internet connection removes the complex telephone setups required by many conferencing tools. Second, the prevalence of the HTML document standard on the Web means that a large amount of existing information can be brought into the meeting. Third, most computer-literate people are now familiar with the interfaces of Web browsers, so little training would be required.

A few browsers today provide rudimentary groupware abilities e.g., NCSA Mosaic and [1]. Typical features include page synchronization and chat windows. Drawing on research into groupware workspaces and shared drawing tools [2], we suggest a set of changes that can turn Web browsers into excellent presentation tools. These are described by way of our implementation of GroupWeb, a browser that allows distributed groups to navigate and share Web pages in real time.

THE GROUPWEB DESIGN

GroupWeb is built on top of GroupKit, a groupware toolkit [3]. Like normal Web browsers, GroupWeb fetches and displays HTML pages. However, several people can enter a GroupWeb session via a session manager (Figure 1f). Each participant runs their own GroupWeb replica, and the browser becomes a shared visual workspace (Figure 1a). Each replica has independent access to the Web, but they communicate directly to each other to stay synchronized.

GroupWeb is founded on five design requirements: document slaving for synchronizing pages, relaxed "what you see is what I see" views to handle display differences, view slaving for synchronous scrolling within pages, telepointers for enacting gestures, and group annotations that can be attached to pages. Each are described below.

Document slaving for synchronizing pages

The most basic act of a presenter is selecting material (a slide or page) and bringing it to the group's attention. In GroupWeb, the material is an HTML page that the presenter selects by navigating a link. GroupWeb guarantees that all the browsers in the session receive a copy of the page, which we call document slaving. It does this by instructing all its replicas to fetch the new page, specified by the HTTP address.

Relaxed "what you see is what I see" views

In face to face meetings, all people see exactly the same thing. In distributed meetings, a similar effect can be achieved by strict "what-you-see-is-what-I-see" (or strict-WYSIWIS), where the visuals are kept identical across all displays. However, display sizes and personal desires differ, and it may be onerous to impose this constraint onto distributed meeting participants. GroupWeb "relaxes" strict-WYSIWIS by permitting windows to be different sizes, and by reformatting the text to fit the display nicely. While this means that people may not see exactly the same thing, it does provide more flexibility for the way each person wishes to view their page.

View slaving for synchronous scrolling

Most HTML pages require scrolling as they rarely fit completely within a window. In a shared document, scrolling can be independent or synchronized. GroupWeb allows both. Independent scrolling, a form of relaxed-

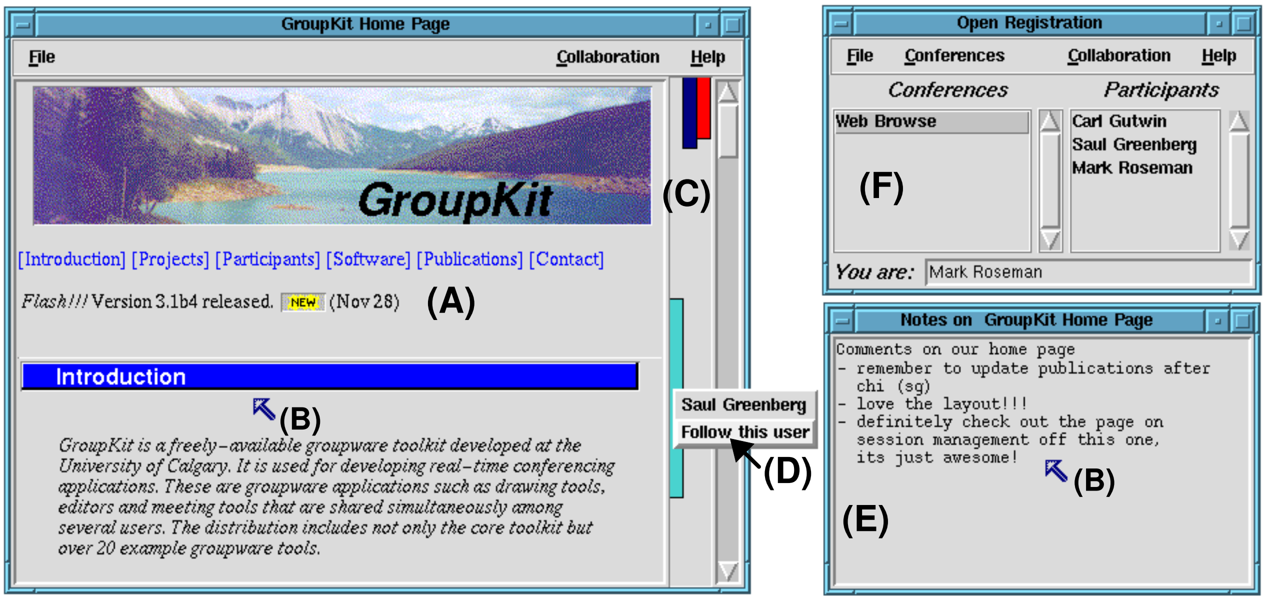

Figure 1. GroupWeb, showing (a) a shared web page, (b) telepointers, (c) multi-user scrollbars, (d) view-slaving controls, (e) a group annotation tool, and (f) a session manager for joining the GroupWeb conference.

WYSIWIS appropriate for loosely coupled collaborations, is the default. People can have viewports onto different parts of the page, with feedback on other's locations shown via multi-user scrollbars (Figure 1c). Each colored bar represents how much a particular participant can see, as well as the overlap between views, if any. Synchronized scrolling, on the other hand, automatically aligns viewports. To enable this, a menu on each bar is raised (Figure 1d), with the menu (and the bar's color) identifying the participant it represents. Selecting "Follow this user" causes the local display to scroll in synchrony with the other. Because synchronization is one-way (unless the other person also slaves the view), the local user can still scroll quickly to other parts of the page for quick glances.

Gesturing through telepointers

Hand gestures play an important role in any work surface activity, and presentations are no exception. We use them to enact ideas, to signal turn-taking, to focus attention of the group, and to reference objects on the work surface [2]. As with many groupware workspaces, GroupWeb uses telepointers as a way to transmit and display gestures (Figure 1b). Because the display is relaxed-WYSIWIS, telepointers are attached to letter positions rather than Cartesian coordinates. This means that the pointer will always be over the same text on all displays.

Attaching group annotations to pages

While the original web document is not editable, people can attach shared annotations to any page. GroupWeb includes an annotation tool (Figure 1e), which is a multi-user text editor. Users can enter and edit text at any time, which is displayed on all screens. As well, the annotation is automatically keyed to the current web page. Changing to a new page clears the editor, while returning to an annotated page restores the text annotation in the editor. Annotations can be used for almost anything: group note-taking, collecting comments, suggesting revisions, and chatting.

DISCUSSION AND SUMMARY

We have described a system called GroupWeb which leverages the ubiquity of the Web by adding features found in real-time groupware systems. This combination can yield a powerful presentation tool that could be used for many purposes, including distance education, distributed meetings, and small group collaboration.

Groupware browsers, however, will have to evolve along with the capabilities of Web pages. Page synchronization works now because current pages contain mostly static information. With the recent introduction of both forms and the Java language, pages will have to synchronize themselves at the input event level (for forms) and script execution level (for Java) as well.

REFERENCES

1. Dybvik, P. and Lie, H. (1994) Combining real time multimedia conferencing with hypertext archives in distance education. *ACM SIGRAPH Video Review*, **106**.

2. Greenberg, S., Hayne, S. and Rada, R. (1995) *Groupware for real time drawing: A designer's guide.* McGraw-Hill, Europe.

3. Roseman, M. and Greenberg, S. (1996) Building real time groupware with GroupKit, a groupware toolkit. *ACM Trans Computer Human Interaction*, in press.

Children's Collaboration Styles in a Newtonian MicroWorld

Andy Cockburn
Department of Computer Science
University of Canterbury, Christchurch, N.Z.
Tel: +64-3-364-2774
E-mail: andy@cosc.canterbury.ac.nz

Saul Greenberg
Department of Computer Science
University of Calgary, Calgary, Canada
Tel: +1-403-220-6087
E-mail: saul@cpsc.ucalgary.ca

ABSTRACT

TurboTurtle is a animated multi-user microworld that children use to explore concepts in Newtonian physics. It is a groupware system where students, each on their own computer, can simultaneous control the microworld and gesture in a shared view. Observations of pairs of young children using TurboTurtle highlight extremes in collaboration styles, from conflict to smooth interaction.

Keywords

Computer supported collaborative learning, microworlds.

INTRODUCTION

Microworlds [3] are computer simulations of restricted environments that promote exploratory learning by children. TurboTurtle (Figure 1) is a microworld that simulates a Newtonian universe [1]. Students explore physical concepts by adjusting properties such as gravity, friction, force, and velocity. They immediately see the effects of these changes on the behavior of a turtle (a ball) that moves through the world. What makes TurboTurtle intriguing is that it is group-aware. Small co-located or distributed groups can talk about the simulation while they are manipulating it. Each student has their own computer screen and input devices. They share the same view of the simulation, have telepointers to facilitate gesturing, and can simultaneously manipulate any aspect of the microworld [1].

We wanted to see how children managed, or mismanaged, their collaboration in this environment that not only allowed parallel activity, but that made no attempt to structure turn-taking or mediate conflicting actions.

METHOD

Twelve children, aged ten or eleven, used the system in mixed sex pairs for 30 minutes. They were observed through think-aloud and constructive interaction techniques. Children were seated approximately two meters apart with a clear view of each other. They were assigned tasks that familiarized them with TurboTurtle as a collaborative tool, and that progressively introduced the Newtonian concepts of friction, gravity, mass and force.

OBSERVATIONS AND RESULTS

The children had fun. None left their machine to work directly with their partner on a single machine. Eye contact was rare, but during breakdowns it was common for one child to quickly glance at their partner, without reciprocation. These observations indicate the overall success of TurboTurtle as a shared microworld.

Collaboration styles varied greatly. The summary in Table 1 shows that different pairs talked to each other in quite different ways, and that they manipulated the microworld using various collaboration styles. These are described next.

Collaboration styles

Parallel activity. Pairs two and five continually discussed their actions and managed their collaborations simultaneously and successfully. As part of this, they were vocal about the undesired actions of their partners. For instance, the boy in pair 2 closed off a rocket control without prior warning. The girl turned to the boy and scolded him with "You aborted the mission!"

Sequential activity. Pair three negotiated control to the near exclusion of simultaneous activity. For example, when asked to set the rocket controls, the girl said "You set the heading, then I'll set the fuel and force. Then you can launch the rocket." The sequence was carried out in that order with no overlapping of actions, and with an explicit "OK" once each stage was completed.

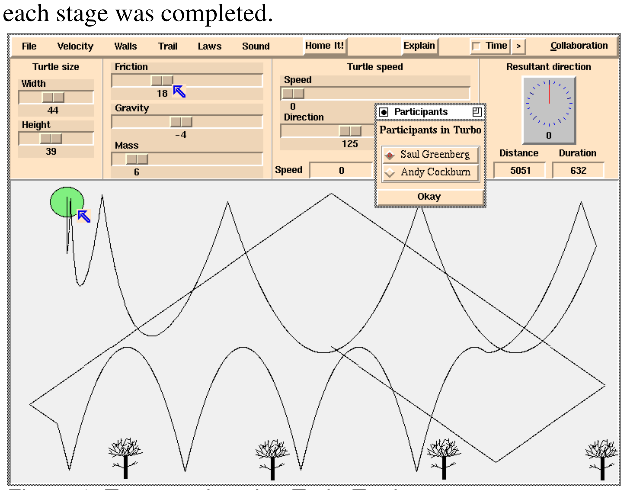

Figure 1. Two people using TurboTurtle.

	Sex	Individual style	Collaboration style
1	boy	rapid speech and manipulation	Boy dominated the collaboration with continuous fast speech and rapid manipulation of the microworld. Girl almost totally excluded except when invited to do something by the boy.
	girl	little speech or manipulation	
2	boy	continuous discussion	Fluid and dynamic shared control of the microworld. Periodic breakdowns over task aspects, with appropriate admonishment.
	girl	continuous discussion	
3	boy	conversation after breakdown	Mostly sequential interaction. Extensive negotiation over the management and ordering of activities, with the girl taking the leading role
	girl	continuous 'think aloud'	
4	boy	continuous speech	The boy primarily drove the collaboration, with continuous invitations for the girl to carry out activities.
	girl	continuous speech	
5	both	continuous discussion	Good shared control of the microworld.
6	boy	almost no speech	Very poor use of the microworld, resembling single user use. They almost ignored the fact that they were working together.
	girl	almost no speech	

Table 1: Predominant collaboration styles for six pairs of children, aged 10-11 years old.

Independent activity. Pair six almost ignored the fact that they were in a collaborative microworld. They were mostly silent despite being encouraged to communicate. They struggled against the actions of each other, even though the telepointers revealed the cause of their difficulty. They said it would be much easier to use the microworld on their own.

Domination. Breakdowns also occurred when one person dominated the interaction. In pair one, the boy changed the simulation properties so rapidly that the girl could not keep pace. The girl initially took her hands away from the mouse, clearly attempting to follow the frenetic activity of her partner. Shortly afterwards she shouted "Leave it!" While the boy briefly capitulated, he continued to dominate the session, grabbing the controls whenever the girl hesitated.

Breakdowns

Breakdowns happened even in successful collaborations. Yet many were positive contributions to the overall interaction, with the breakdown becoming a focal point for children negotiating their next manipulation of the microworld. Two factors mitigated breakdown: conversation and mutual awareness.

Conversation. Successful breakdowns were distinguished from unsuccessful ones by the extent of discussion that accompanied the conflict. For instance, pair two argued over the desired mass of the turtle, set by a slider. Their short conflict was accompanied by comments such as "Make it 20!", "No! Make it 30!". Note that the conflict stems from the task, rather than the interface. In contrast, pair six encountered the same problem of simultaneous access to a slider, but it was not clear to them whether the values that they were trying to set were the same or different, the confusion being caused by their total silence.

Mutual awareness. While conflicts over the simultaneous access of sliders were frequent, children were aware of the problem because they saw the two telepointers on the slider as well as the bouncing slider position as both tried to move it. They can then repair the conflict through their natural social skills, much as they do in the real world. This did not always happen. In some cases the children were tenacious in their desire to be last one in control, even though they were well aware of the cause of the problem. This problem arose because of their own immaturity at negotiating control.

Still, the importance of mutual awareness to resolve conflict was emphasized by two bugs. First, in one part of TurboTurtle telepointers are not visible. Frustrated comments such as "Hey, how did that happen," and "What are you doing?" were frequent. Next, some parts of TurboTurtle did not show exactly the same view e.g., pull-down menu actions are not shared, and two pairs said they wanted to be able to see their partner's menu selections. These small breakdowns indicate the importance of keeping aware of another's activities in the microworld.

DISCUSSION AND SUMMARY

Collaborations worked when children negotiated their interactions, and used mutual awareness and breakdowns to further their discussions. They were less effective when children dominated one another, ignored each other, or fought for control of microworld objects. Yet unlike other microworlds, these problems did not arise because children shared a single input device and display. We agree with Cole [2], who interprets children's control of their collaborations in the microworld to be a social process developed through their own group dynamics. The implication is that groupware microworlds should give children both the freedom to explore the simulation at their own pace and personal style, while adding appropriate structure to minimize the risk of detrimental breakdown that occurs because children are immature collaborators.

REFERENCES

1. Cockburn, A. and Greenberg, S. (1995) TurboTurtle: A collaborative microworld for exploring Newtonian physics. *Proc Computer Supported Collaborative Learning*, pp. 32-39, Oct 17-20, LEA Press.
2. Cole, K. (1995) Equality issues in computer based collaborations: Looking beyond surface indicators. *Proc Computer Supported Collaborative Learning*, pp. 64-74, Oct 17-20, LEA Press.
3. Papert, S. (1993) *The children's machine: Rethinking school in the age of the computer.* Basic Books.

TeamRooms: Groupware for Shared Electronic Spaces

Mark Roseman and Saul Greenberg

Dept. of Computer Science, University of Calgary
Calgary, Alta, Canada T2N 1N4
Tel: +1-403-220-3532
E-mail: roseman, saul@cpsc.ucalgary.ca

ABSTRACT

Teams whose members are in close physical proximity often rely on team rooms to serve both as meeting places and repositories of the documents and artifacts that support the team's projects. TeamRooms is a prototype groupware system designed to fill the role of a team room for groups whose members can work both co-located and at a distance. Facilities in TeamRooms allow team members to collaborate either in real-time or asynchronously, and to customize their shared electronic space to suit their needs.

KEYWORDS: Groupware, CSCW, shared electronic spaces

INTRODUCTION

Johansen et. al. [2] describe how team rooms have become a central device used by business teams to organize their work. Team rooms provide a permanent shared space used by the team, serving as a meeting room, work area, a place to store documents that are needed by the team's projects, and more generally, as a focus for communication within the group.

Traditional team rooms rely on the physical proximity of the team members and their easy access to the room. We describe here a prototype groupware system called TeamRooms that supports the team room concept for teams whose members work either co-located or at a distance. It combines aspects of both real-time and asynchronous groupware to provide the team with a shared electronic space. The system is highly customizable, allowing the team to design their electronic room on the fly to suit their needs, as they do with their physical meeting rooms.

Users run a TeamRooms client that connects over a network to a server providing a number of rooms. Each room contains both generic communication tools (a chat tool and a backdrop acting as a shared whiteboard) and any number of *applets* needed to support the group's work. Typical applets would be diagramming tools, outliners, brainstorming tools, browsers for information such as web pages, notes to other team members, as well as more frivolous items such as card games. When team members are in a room at the same time, they see each other's actions both through changes in the room's artifacts and through mechanisms such as multiple telepointers. As with real rooms, all artifacts in the electronic room persist even when no one is in the room.

TeamRooms is implemented using our groupware toolkit GroupKit [3], augmented to support centralized processes, user authentication, a versioned persistence repository, and embedded conferences. Each applet is built as a standard GroupKit application, which allowed us to easily move a number of existing applications into TeamRooms, and to rapidly create new ones.

TEAMROOMS INTERFACE

Figure 1 illustrates the user interface of the TeamRooms client, where the user (Carl) is in a room called "Ideas for Papers" with two other users (Saul and Mark). Along the bottom of the screen are a text-based chat tool and different colored pens for drawing on the "walls" of the room (a shared whiteboard). Also shown are three applets: a group outliner, a sticky note, and a URL pointer.

Each applet is embedded in its own frame, in a similar fashion as Opendoc or OLE components. Users select new applets from the Tools menu, as well as delete, move and resize them. All changes are immediately visible to all users in the room. TeamRooms also keeps a complete version history for each applet, allowing users to retrieve an earlier version. This creates a new instance of the applet in the room, allowing comparison between versions.

Interestingly, TeamRooms has led us to construct several applets that we would never have considered in conventional groupware. For example, one tool allows inserting a pointer to outside information (a URL, which can be changed by any user), which when selected will invoke a user's World Wide Web browser. Another tool contains a pointer (again as a URL) to an image to be displayed in the room. This allows

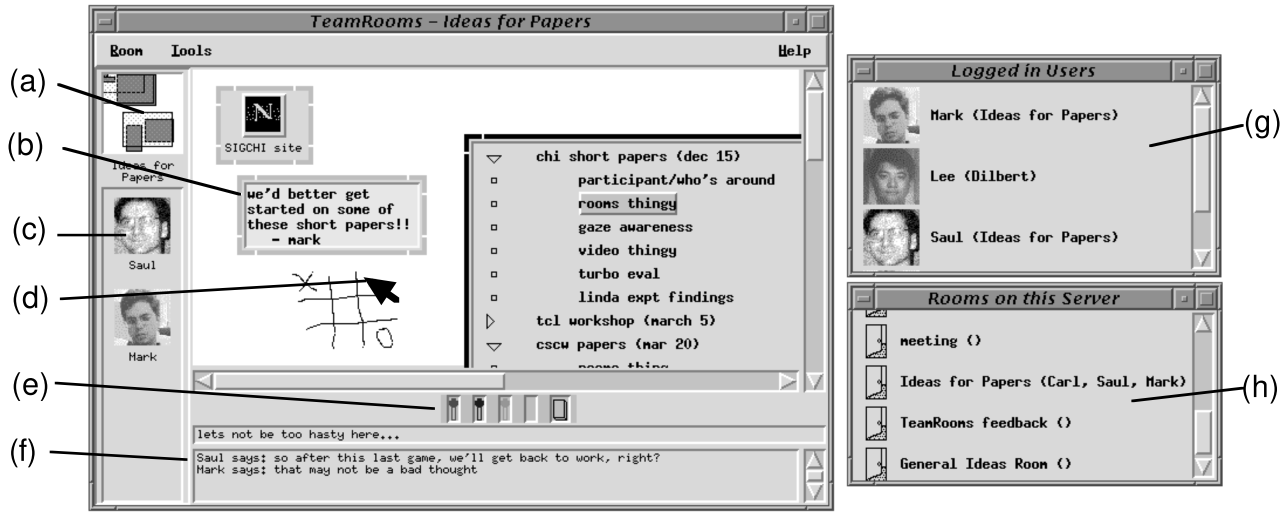

Figure 1. TeamRooms User Interface, showing (a) radar view of room; (b) applets; (c) image stills of users in room; (d) telepointer; (e) whiteboard pens; (f) text-chat area; (g) list of users currently around; (h) list of rooms.

a team to create a room by importing or referring to relevant outside information (or at least information of interest; our most common image was the daily Dilbert comic).

TeamRooms also provides several facilities for maintaining awareness of other team members. The two windows on the right of Figure 1 show who is around and in which rooms they are working. Within the current room, its participants are always visible. Telepointers show users' fine-grained activities. Because rooms can be larger than a user's window, the radar view shows a miniature of the room, indicating the position of applets (dark boxes) as well as other users' views into the room (light stippled boxes).

RELATED WORK

Conventional real-time groupware systems often focus on a "meeting" or "session". Such systems support distributed teams through isolated activities (e.g. a shared whiteboard session), but provide limited and heavyweight support for other team activities. In contrast, asynchronous groupware (such as workflow or email) often does not support real-time activities. Neither are successful at providing the overall collaboration support offered by conventional team rooms.

A popular class of systems that parallel the team rooms idea are Multi-User Dungeons (MUDs). Traditional MUDs are text-based systems where users connect to a central server. Once connected, they enter any number of different rooms, chat with other users in those rooms, and create and modify artifacts in the rooms. Though primarily used socially, MUDs have been used to support collaborative work, though their text-only presentation has proved limiting.

We are not unique in trying to provide more sophisticated media in shared electronic spaces. The Jupiter system [1] augments a traditional MUD (LambdaMOO) with audio/video conferencing tools and shared whiteboards. The wOrlds system [4] also supports audio/video conferencing as well as numerous other tools. TeamRooms is most similar to wOrlds, but is more lightweight, does not directly support audio or video (though it could be extended to do so), and emphasizes applets more suitable for real-time collaboration.

CONCLUSIONS

TeamRooms is a prototype groupware system combining the rich applications and interfaces of real-time groupware tools with the persistence and work context provided by MUDs. The result is a system providing the electronic equivalent of a team room for co-located or distributed work groups. Our early usage experiences with TeamRooms are quite encouraging, and we have found that it does afford many of the same behaviors seen when teams share a physical space.

REFERENCES

1. Curtis, P. and Nichols, D. MUDs Grow Up: Social Virtual Reality in the Real World. In *Proc. of the Third International Conference on Cyberspace*. May 1993.

2. Johansen, R., Sibbet, D., Benson, S., Martin, A., Mittman, R. & Saffo, P. *Leading Business Teams*. Addison-Wesley. 1991.

3. Roseman, M. and Greenberg, S. Building Real Time Groupware with GroupKit, a Groupware Toolkit. *ACM TOCHI* (1996, in press).

4. Tolone, W., Kaplan, S. and Fitzpatrick, G. Specifying Dynamic Support for Collaborative Work within wOrlds. In *Proc. of ACM COOCS*. 1995.