Greenberg, S. (2002). Real Time Distributed Collaboration. Grouplab Report, Dept. Computer Science, University of Calgary.

Real Time Distributed Collaboration

Saul Greenberg

Department of Computer Science
University of Calgary
Calgary, Alberta
Canada T2N 1N4
saul@cpsc.ucalgary.ca
+1 403-220-6087

Abstract

Groupware for real time distributed collaboration allows people to work together at the same time, even when some or all participants and their work products are in different physical locations. To do this effectively, groupware and its components must support *telepresence*—a way of giving participants enough cues about each other to help them orchestrate their interactions—and *teledata*, a way of having participants bring into the meeting the materials and on-going work they wish to share with one another. Consequently, the design and implementation of a distributed system supporting real time collaboration must handle the human factors of how people collaborate as well as the expected technical issues. This article will describe some of these human factors, and how they are addressed by several groupware applications.[1]

Telepresence

In face to face conversation and collaboration, people use and rely on many subtle cues to mediate their activities. These include voice inflection and pauses, body language, hand gestures, eye contact, gaze awareness (i.e., knowing where others are looking), and so on. In turn, these cues are used for many purposes: knowing who is speaking and who is listening, mediating turn-taking, focusing attention, detecting and repairing conversational breakdown, and building a common ground of joint knowledge and activities. (Clark [1] provides a comprehensive discussion of the role of these cues in everyday language.)

The goal of supporting telepresence in real-time distributed groupware is to capture and transmit both the explicit and subtle dynamics that occur between collaborating participants. This is no easy task. For example, traditional voice and video conferencing systems capture only a small part of these dynamics. When the voice channel is of low audio quality, a person's speech dynamics are not as clear to others as they could be. When voice is non-directional, participants of multi-point conferences find it difficult to associate a voice with a particular speaker. With half-duplex channels, people cannot speak at the same time. This also makes it is harder for listeners to interrupt or to inject back-channel utterances such as 'ok', 'ums', and so on. Video channels are also problematic. Because of camera positioning, a participant who is looking directly at another person's eyes in a video image is in turn seen as staring at their navel: this happens when the video

[1] This is an invited article solicited for an Encyclopedia of Distributed Computing. However, the encyclopedia never made it to press.

camera is mounted above the video monitor. When compressed video is used to preserve bandwidth, the jerky and often blurred image looses almost all the subtle body cues. Even with full video, the tightly zoomed in ‘talking head’ means that body gestures are not visible. Yet zooming out to include the whole body compromises image fidelity and resolution. (The edited collection by Finn, Sellen, and Wilbur [2] includes excellent discussions on the opportunities and limitations of video-mediated communication.)

Research is now addressing the human factors of telepresence. Eye contact in video can be maintained just by positioning cameras and screens behind half-silvered mirrors. Systems can even mimic people’s spatial relationships to one another in a multi-point collaboration, allowing people to turn to and speak to one another just as they do in real life. To do this, the system projects people’s images and voices onto separate video monitors and speakers, whose relative positions are equivalent in all locations. An impressive realization of this configuration occurred in the early 1980’s, where researchers at MIT created physical models of ‘talking heads’. They fashioned a transparent mask of a participant, mounted it on a motorized platform at the remote site, and then projected the video image of the participant into the mask. Through sensors, the mask would move to reflect the person’s actual head movement. The result was striking, where the talking (but disembodied) head was so realistic that it proved disturbing to its viewers! More recently, virtual reality environments offer telepresence through avatars: synthetic bodies that populate a 3-d landscape. While most avatars are extremely crude, some systems attach a person’s video image to the avatar, transmit hand and body gestures, and indicate where a person is looking in the environment.

In summary, the human factors of telepresence demands careful attention. The naïve view is that telepresence can be supported by low quality video and audio. The reality is that even the best research systems only supply partial telepresence. (*See also* Computer supported cooperative work, Group communication, and Videoconferencing.)

Teledata

Teledata brings to the distributed meeting work materials, such as notes, documents, plans and drawings, as well as some common work surface that allows each person to annotate, draw, brainstorm, record, and convey ideas during the meeting's progress. (The edited collection by Greenberg, Hayne and Rada [3] provides case studies of the design, implementation and application of systems supporting teledata.)

Teledata is usually implemented in one of three ways.

1. *A video-based system* captures the work surface and the objects within it as a video image. In simple systems, a camera is just positioned over the work area and its image transmitted. This presents only a one-way view of another’s work area. Alternatively, video can provide participants with a common workspace by fusing two work areas together into a single image via technology that includes video overlays. The restriction is that while people can see each other’s objects in the common image, they cannot manipulate the objects held by the other people.
2. *A view sharing system,* also known as a *collaboration-transparent* system, takes a standard, unaltered single-user computer application and displays it on the screen of all participants. Each participant sees the same image on their display and can interact with it by taking turns. This is similar to several people sharing a single computer; each sees the same thing and can pass the

keyboard around, but the application has no idea that it is being used by more than one person. The restrictions are that simultaneous activity is not possible, and that the software being shared is not designed to handle the subtleties of group interaction.

3. *Collaboration-aware groupware* is specifically designed with the group in mind. The software knows that there are a number of people interacting with it, treats each participant's input separately, and may customize the view presented on each person's display. These systems are tailored to group needs, and most allow simultaneous activity.

As with telepresence, systems for teledata must support the human factors of group interaction. Most shared visual workspaces have many characteristics in common [3]. First, people *manipulate* objects in the space; they create objects, move them around, modify them and remove them. The implication is that the space must be an interactive one. Second, people often *gesture* around a workspace to communicate specific information to others. Gestures are often tied to speech, such as when one person points to an object and says "this thing over here". Thus systems should support a person's ability to talk and gesture around the workspace. Third, people use the workspace as a medium to *express ideas* to one another, where they talk as they manipulate objects. Thus object manipulation must be visible at all sites with no apparent delay if they are to act as conversational artifacts. Fourth, people shift between *loose and tight collaboration* over time, where they move constantly and fluidly between individual and group work. This means that people should be able to focus their attention on different parts of the workspace when they are doing individual work. Fifth, people maintain *awareness* of what others are doing as they are doing it. Thus the workspace must provide enough information to let people know who is in the workspace and where they are working, what they are doing, and what changes they are making [5].

The collaboration-aware groupware system in Figure 1 illustrates how these and other human factors can be incorporated into a design [5]. The system is a groupware concept map editor, where groups can create and organize ideas by manipulating nodes and arcs. Most of the window in Figure 1 shows a participant's "detailed view", which is a viewport into a portion of the shared work surface. At the top left corner of the figure is a "radar view", which shows a miniature overview of the entire work surface. The system supports the following group activities.

1. People can simultaneously manipulate the objects in either the radar or detailed view.
2. People gesture around the workspace through *telepointers* (large arrows), one for each participant, that act as surrogates for people's hands. People can point to objects in both the radar view and the detailed view.
3. People can express ideas, for all fine-grained actions are visible as they happen. When participants are not looking at the same thing in the detailed view, they can see other people's actions in the radar view.
4. People can pursue individual as well as collaborative work. For more individual work, they can scroll to different parts of the concept map by moving their view rectangle (the colored box) in the radar view. If they want to work closely together, they can quickly align their view rectangle atop one another.
5. People can maintain awareness of one another even when working on quite different parts of the concept map. The radar view shows other people's presence and location through both the view rectangles and the telepointers, and can see what others are doing because all actions are immediately visible. The detailed view also shows where another person's view overlaps with one's own by displaying the common region as a colored box.

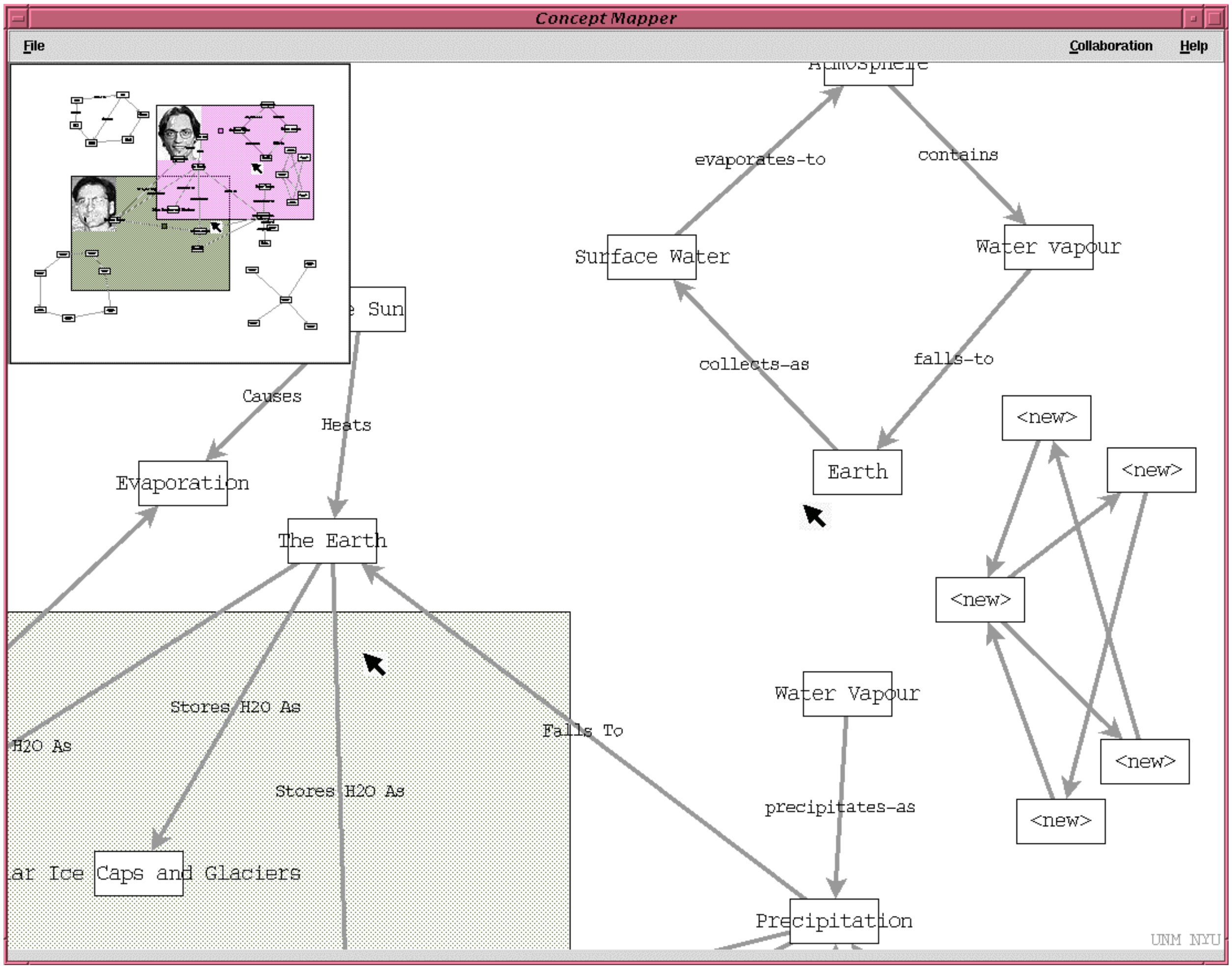

Figure 1. A concept map groupware application, showing a radar overview in the upper left corner and telepointers (from [5], with permission).

Of course, telepresence and teledata should work together. Most teledata systems only support a partial sense of telepresence (e.g., the telepointers in the system above). One notable exception was produced by Ishii and Kobayashi from NTT [6] in their ClearBoard 2 system (Figure 2).

Telepresence is through video images of people's bodies and hands, which are captured, transmitted and displayed through technology that includes video projection, cameras, half-silvered mirrors and polarizing film. Teledata is through a transparent digitizing sheet that is part of the ClearBoard screen. The digitizing sheet runs a groupware drawing system, and participants can interact with it through a digitizing pen. The result is a fluid integration of presence and data. The configuration of the cameras and display means that people maintain a strong notion of gaze awareness, where one can tell exactly where the other is looking. This includes eye contact and eye gaze. The video capture of people's hands means that all gestures are transmitted relative to the workspace objects. The consequence is that people feel as if they are "looking *through* and drawing *on* a big glass board", a metaphor that can be easily understood by all users. (*See also* Computer supported cooperative work, Videoconferencing, Decision support systems, Multi-player games).

Figure 2. ClearBoard-2 in use. From [6], with permission.

Other considerations

This article has just scratched the surface of real time collaboration. There are a variety of other human and technical issues that must be attended to. A sampling is listed below.

Groupware widgets. Perhaps the greatest benefit of today's graphical user interface toolkits is their provision of tried and tested interface widgets that programmers can configure and position in a few lines of code. When widgets are designed by interface experts, the everyday programmer can insert them into the application with some assurance that they are usable. Groupware programmers have the same need for widgets of value to conference participants. However, groupware widgets differ from normal widgets. They have different semantics, actions performed on them must be reflected across displays, and novel widgets have to be designed that address needs specific to groupware [4].

Session management lets people control and establish their groupware connections, meetings, and encounters with others. Session managers are often presented through metaphors. A telephone metaphor, for example, implies that people “call” one another to initiate a groupware session. A spatial metaphor means that people can navigate a space, see who is around in it, and initiate conversation with people they meet. A rooms metaphor extends the spatial metaphor by providing rooms containing persistent groupware applications. As people enter a virtual room, all connections between people and between their groupware applications are automatically made.

Concurrency control is required to guard against inconsistencies, and to handle conflicting actions. However, concurrency control in groupware must be handled differently than traditional concurrency control methods, simply because the user is an active part of the process. For example, people doing highly interactive activities will not tolerate delays introduced by conservative locking and serialization schemes. Similarly, they must be able to understand the effects of any undo/redo and transformation mechanisms required to repair inconsistencies in optimistic schemes. Finally, people can manage certain types of conflicting actions by social rather than technical means, implying that some indication of conflicts must be shown within the interface. (*See* Concurrency control*)*.

Access control determines who can access a groupware object and when. Access control may be required when people wish to have their own private objects, where only they can manipulate or view them. While access control is well known in distributed systems research, the human factors of groupware implies that it be managed in a light-weight, fine-grained fashion. If it is not, it will intrude onto the interface, where people must fight with the system to move between notions of public and private objects. (*See* Access control).

Security and privacy. Groupware could be a large security hole unless great care is taken in determining that only the right people are allowed in a meeting, and that transmissions are private. Because groupware executes actions at many sites, participants need assurances and safeguards that the groupware will not compromise their local system's integrity.

Fault tolerance. Groupware applications should degrade gracefully. They should make reasonable decisions on how quality of service is affected, checkpoint failed conferences for later resumption, and seek alternate communication paths when a channel is no longer adequate.

Groupware toolkits. A groupware toolkit provides programmers with both development tools and a run-time architecture. If it is well designed, the toolkit will automatically provide many of the features described above.

In summary, groupware for real time collaboration requires careful attention to both technical and human factors. The human factors should drive the design, for there are many requirements and nuances that determine whether a system will support collaboration effectively. This implies that the technical approaches to distributed systems, such as those described in this encyclopedia, must be reconsidered to see how they match the actual needs of people.

References

[1] Clark, H. (1996) Using language. Cambridge University Press. ISBN 0-521-56745-9.

[2] Finn, K., Sellen, A., and Wilbur, S. (Editors) (1997). *Video-mediated communication.* LEA Press. ISBN 0-8058-2287-8.

[3] Greenberg, S., Hayne, S. and Rada, R. (Editors) (1995) *Groupware for real time drawing: A designer's guide.* McGraw Hill. ISBN 0-07-707899-3.

[4] Greenberg S. and Roseman M.(1998). *Groupware Toolkits for Synchronous Work.* in M. Beaudouin-Lafon (ed.), *Trends in CSCW*, John Wiley & Sons Ltd.

[5] Gutwin, C. (1997) *Workspace Awareness in Real-Time Distributed Groupware*. PhD Dissertation, Department of Computer Science, University of Calgary, Canada. Available at: www.cpsc.ucalgary.ca/grouplab/papers/

[6] Ishii, H. and Kobayashi, M. (1992) ClearBoard: A seamless medium for shared drawing and conversation with eye contact. in *Proceedings of the ACM CHI'92 Conference on Human Factors in Computing Systems*, p525--532, Monterey, California, May 3--7. Reprinted in [4, Chapter 6],