# GroupSketch: A multi-user sketchpad for geographically-distributed small groups

Saul Greenberg
Department of Computer Science, University of Calgary, Calgary, Canada
email: saul@cpsc.ucalgary.ca

Ralph Bohnet
MPR TelTech Ltd, 8999 Nelson Way, Burnaby, Canada

## Abstract

Recent research observations of small group meetings have identified factors critical to the design of computer tools supporting real time collaborative design. In particular, group activity revolves as much around the design process—sketching, annotating, listing ideas, and gesturing around a communal work surface—as it does around the resulting drawing artifact. A workstation-based tool called *GroupSketch* has been developed that allows a small geographically-distributed group to list, draw, and gesture simultaneously in a communal work surface, supporting interactions similar to those occurring in the face-to-face process. *GroupSketch* facilitates collaboration by: a) allowing gestural expression through large unique cursors visible on all displays; b) minimizing overhead encountered in storing information; c) conveying the process of expressing ideas by transmitting small granular changes of user activity with minimum time delay; d) intermixing gestural, textual, and graphical expression modelessly; and e) providing simultaneous access to a common view of the work surface area. Observations drawn from actual design sessions indicate that people use *GroupSketch* in much the same way they use face to face communal sketchpads.

**Keywords:** shared workspace, real time remote conferencing, computer supported cooperative work, groupware, human computer interaction.

## 1 Introduction

Although computers are now familiar tools used by people to pursue their own individual tasks, they have not, until recently, been exploited to assist people working together. *Groupware* is software that explicitly supports group work (eg Johansen 1988). It is a technically-oriented label meant to differentiate "group-oriented" products explicitly designed to assist groups of people working together from "single-user" products that help people pursue their isolated tasks. *Computer-supported cooperative work* (CSCW) is the scientific discipline that motivates and validates groupware design (eg Greif 1988; Galegher, Kraut and Egido 1990; Greenberg 1991a). It is the study and theory of how people work together, and how the computer and related technologies can or does affect group behaviour[1].

This paper will limit itself to real-time geographically distributed conferences that bring people together in formal or semi-formal meetings, even when some or all participants are physically distributed over different locations (Greenberg and Chang 1989). Most research efforts in distributed conferencing have been in the field of *tele-presence*—a way of giving distributed participants a feeling that they are in the same meeting room (Egido 1988; Johansen and Bullen 1984; MIT 1983). The goal of tele-presence is to transmit both the explicit and subtle dynamics that occur between participants. These include body language, hand gestures, eye contact, meta-level communication cues, knowing who is speaking and who is listening, voice cues, focusing attention, and so on. Tele-presence facilitates effective management and orchestration of remote meetings by the natural and practised techniques used in face to face meetings. *Tele-data*, on the other hand, allows participants at a meeting to present or access physical materials that would normally be inaccessible to the distributed group (Greenberg and Chang 1989). These include 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. Given that an individual's work is commonly centered around a workstation, the networked computer can become a valuable medium for people to share on-line work with each other.

This document will focus on a groupware system called *GroupSketch*, a multi-user sketchpad supporting remote design activities by small groups. It begins with a summary of CSCW studies of face to face design teams and the resulting design principles generated from them. We then explain how these principles were instantiated in the groupware implementation, and briefly describe the underlying architecture. Observations of *GroupSketch* in use follows. The final section contrasts *GroupSketch* to related groupware efforts.

[1]See Greenberg 1991b for a bibliography of CSCW.

## 2 Small Group Design Meetings

Almost every group process begins with a set of initial design meetings, where participants express, discuss, and develop ideas. It is a creative forum where people are encouraged to present their thoughts to the group, to build upon the ideas presented by fellow members, and to problem-solve. Participants typically use some large communal *work surface*—a group drawing area—to facilitate their interactions. Examples include whiteboards, flipcharts, large sheets of paper, as well as a variety of coloured pens for drawing.

Physical work surfaces used by face to face meetings are rather limited. Consider, for example, the pervasive whiteboard. Although it is an excellent and quite general medium for providing a shared and focused memory for a meeting, a whiteboard is physically restricted (Stefik, Bobrow, Foster et al. 1987a; Stefik, Foster, Bobrow et al. 1987b). The amount of surface available for drawing is fixed. Only a few people can simultaneously use it. Whiteboards are ill-suited for re-arranging existing items. They cannot normally keep a record of the artifacts drawn, nor do they allow previously prepared information to be imported easily. They are useless for tele-data as the image is, of course, only visible at one physical site. Given the trade off between capabilities and limitations of this and other physical media, the computer has potential to include the best offerings of existing work surfaces while limiting physical restrictions, and the potential to extend what these surfaces currently enable for meeting participants.

**2.1 Understand collaboration.** In order to design a software-based work surface, we must have an adequate understanding of how traditional ones fit into the meeting process. Indeed, Grudin has identified a lack of understanding of group behaviour to be one of the reasons why groupware has not been generally successful (Grudin 1989). He asserts that designers rely too heavily upon their own intuition, which is often based upon experiences that may not be applicable to the group as a whole.

For example, an intuitive "conventional" view of the communal work surface would consider it merely as a medium for creating and storing a drawing artifact (Tang 1989). Bly disproves this naive view (Bly 1988). She studied two designers communicating through three different media offering different access to a drawing surface: face to face including a shared sketchpad; over a video link that included a view of the other person and their personal drawing surface; and over the telephone. From her observations, she asserts that the the drawing process—the actions, uses, and interactions on the drawing surface—are as important to the effectiveness of the collaboration as the final artifact produced. Bly also noticed that allowing designers to share drawing space activities increases their attention and involvement in the design task. When interaction over the drawing surface is reduced, the quality of the collaboration decreases.

Tang refined Bly's findings even further through his ethnographic study of eight short small-team design sessions (Tang 1989; Tang 1991; Tang and Leifer 1988). Each team used large sheets of paper as a shared work surface and were given tasks to design. Some teams placed the paper on a table, others tacked it to a whiteboard. Even this simple difference had a profound effect on how the group used the shared work surface.

*Orientation.* When people sat around the table, drawings made on the table-mounted paper were oriented in different directions. Although people had greater difficulty drawing and perceiving the images, orientation proved a resource for facilitating the meeting. Because drawings faced a particular person, a context and an audience was established. Marks made by participants that were aligned to an image conveyed support and focus. People working on their own image used orientation as a "privacy" boundary until they were ready to call in the group's attention. The group using whiteboard mounted paper did not exhibit these behaviours.

*Proximity.* Tang noticed that when participants were huddled around the table-mounted paper, the sketchpad played a key role in mediating the conversation. This role was lessened in the whiteboard situation where people were seated several feet away.

*Simultaneous Access.* Given good proximity, a high percentage (45—68%) of work surface activity around the tabletop involved simultaneous access to the space by more than one person.

Tang built a descriptive framework to help organize the study of work surface activity, where every user activity was categorized according to what action and function it accomplished, as listed below (Tang 1989).

*Actions:*

- *listing* produces non-spatially located text or alpha-numeric notes;
- *drawing* produces graphical objects, typically a 2-dimensional sketch with spatially located textual annotations;
- *gesturing* is a purposeful body movement that communicates specific information eg pointing to an existing drawing.

*Functions:*

- *storing information* refers to preserving group information in some form for later recall;
- *expressing ideas* involves interactively creating representations of ideas in some tangible form, usually to encourage a group response;
- *mediating interaction* facilitates the collaboration of the group, and includes turn-taking and focusing attention.

Tang's classification of small group activities within this framework revealed that the "conventional" view of work surface activity—storing information by listing and drawing—constitutes only ~25% of all work surface activities. Expressing ideas and mediating interaction comprised the additional ~50% and ~25% respectively. Gesturing, which is often overlooked as a work surface activity, played a prominent role in all work surface actions (about 35% of all actions). For example, participants enacted ideas using gestures to express them. Gestures were used to signal turn-taking and to focus the attention of the group. Information can be cognitively chunked and preserved through gestures.

**2.2 Implications for design of a work surface.** Tang's observations led him to derive six criteria that shared work surfaces should support. The criteria plus a summary of the reasons why each is offered are listed in Table 1 (condensed from Tang 1989).

## 3. A Description of GroupSketch

**3.1 Overview of the user interface.** *GroupSketch* is a simple group sketching tool that allows up to four people to draw on a virtual piece of paper (the screen). Figure 1 displays a typical *GroupSketch* screen with four participants engaged in a design session. The borders enclose a shared work surface where people can draw, enter text, or gesture. Every person also has a labelled cursor. All participants see the same work surface on their display, and every movement of the cursor and change in the drawing is immediately visible on all displays. Each participant is represented by a unique labelled caricature located immediately outside the work surface, representing seating positions around a virtual table. While audio is not directly supported, we expect a full duplex audio channel to be available by other means (eg speaker phones).

Four action modes are supported: pointing, drawing, listing, and erasing (Figure 1). With no mouse buttons or keyboard keys pressed, the cursor portrays the image of a pointing hand (Irene's cursor). To draw freestyle, the user depresses the left mouse button of a three-button mouse, changing the cursor from a hand to a pen (Sam's cursor). The pen-shaped cursor also appears automatically when typing. Pressing the middle mouse button changes the cursor into a large arrow to draw participants' attention (Bruce's cursor). Users can erase graphics or text in the work surface by holding down the right mouse button, which changes the shape of the cursor into an eraser (Wilf's cursor).

The menu on the right of Figure 1 allows a person to privately save an image, retrieve a previously stored image to the group display, clear the public work surface, or leave the collaboration (leaving other participants in the meeting). Menu selections and cursor movements outside the work surface are private and are not broadcast to other workstations. Loading an image or clearing the work surface will have the same affect on all participant's screens.

In a typical *GroupSketch* scenario, participants converse normally. Yet it is not identical to a face to face meeting. People tend to concentrate intently on the group work surface (they cannot see each other), not only for tele-data but for a limited sense of tele-presence.

| Criteria | Reason |
|---|---|
| 1 Provide ways of conveying and supporting gestural communication. Gestures should be clearly visible, and should maintain their relation with objects within the work surface and with voice communication. | • gestures are a prominent action<br>• gestures are typically made in relation to objects on the work surface<br>• gestures must be seen if they are to be useful<br>• gestures are often accompanied by verbal explanation |
| 2 Minimize the overhead encountered when storing information. | • only one person usually records information<br>• other participants should not be blocked from continuing private or group work while information is being stored |
| 3 Convey the process of creating artifacts to express ideas. | • the process of creation is in itself a gesture that communicates information<br>• speech is closely synchronized with the creation process<br>• artifacts in themselves are often meaningless |
| 4 Allow seamless intermixing of work surface actions and functions | • a single action often combines aspects of listing, drawing and gesturing<br>• writing and drawing alternates rapidly<br>• actions often address several functions |
| 5 Enable all participants to share a common view of the work surface while providing simultaneous access and a sense of close proximity to it. | • people do not see the same things when orientation differs<br>• simultaneous activity is prevalent<br>• close proximity to the work surface encourages simultaneous activity |
| 6 Facilitate the participants' natural abilities to coordinate their collaborations | • people are skilled at coordinating communication<br>• we do not understand the coordinating process well enough to mechanize it |

**Table 1. Implications for design of a communal work surface (condensed from Tang 1989).**

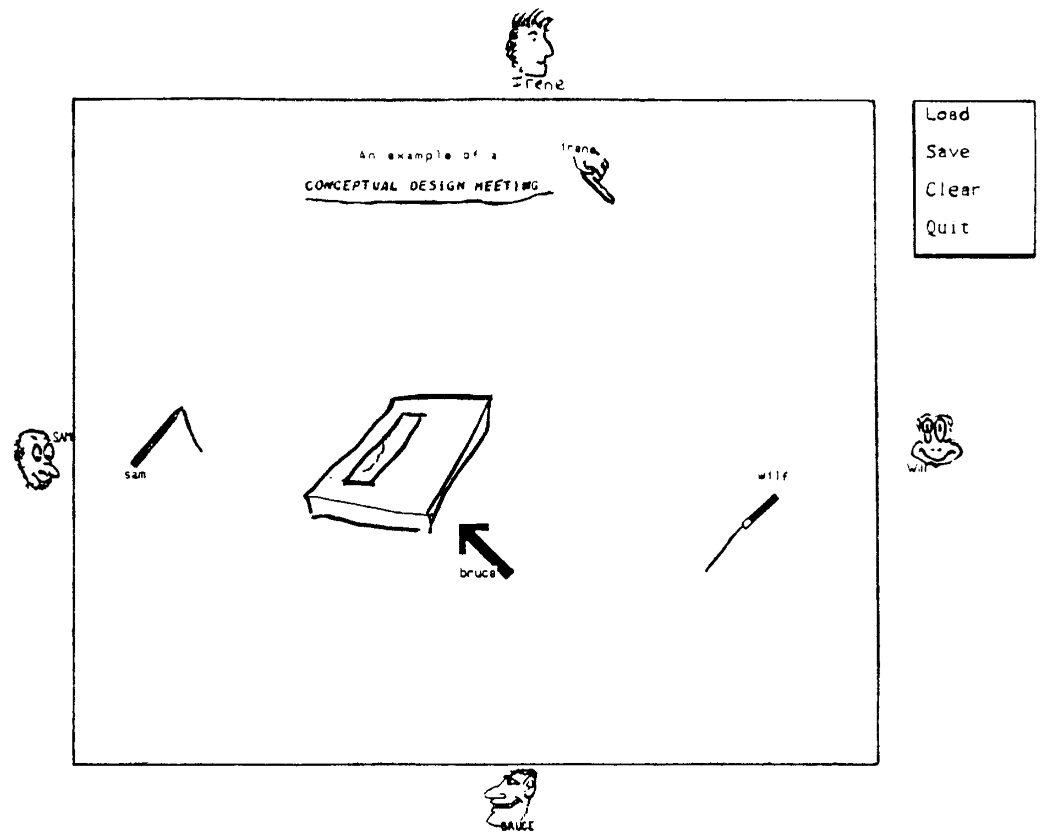



**Figure 1**: A sample *GroupSketch* session

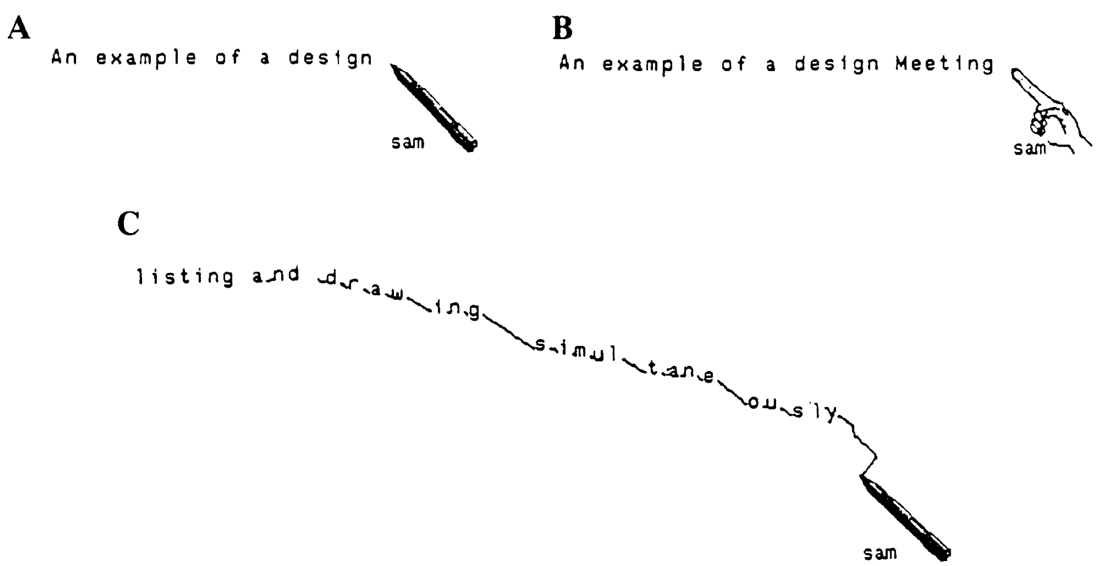


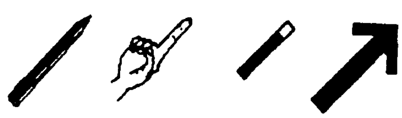

**Figure 2:** Listing and drawing: a) cursor shape while typing and b) when typing is completed; c) intermixing listing and drawing

People focus attention to objects in the display by pointing at them or by circling objects with the cursor. Drawing and listing is both independent (one person responsible for a drawing) and cooperative (multiple people working together on a drawing). People can, of course, work simultaneously on any part of the display, and anyone can be actively gesturing, or creating and editing the drawing artifact.

**3.2 Design considerations.** Despite the simplicity of *GroupSketch*, the design took much consideration, and is directly derived from Tang's criteria listed in Table 1.

*Criterion 1: A collaborative tool should provide ways of conveying and supporting gestural communication.* In *GroupSketch*, the only medium that can convey physical gestures is the computer cursor (augmented by audio). However, there are significant differences between the typical cursor normally available on a workstation (eg single, small cursors) and one we consider suitable for gesturing. *GroupSketch* cursors have the following properties.

- Since gestures must be seen in order to convey information, all cursors within a work surface are always visible to all participants.
- Cursors must have enough prominence on a multi-cursor display to attract the attention of other participants. A large 64x64 bit cursor is used instead of the traditional 16x16 bit cursor.
- Cursors change their shape to reflect a natural action. Four gesture modes are supported (pointing, writing, erasing, and directing attention) by distinct cursor shapes (as shown below).The default cursor shape is the pointing hand. The pen cursor reflects the work surface action of both listing and drawing, while the eraser represents the erasing action. The large arrow allows users to point at and direct the group's attention with greater emphasis than the normal hand.



- Cursors are unique, each identifying the person it belongs to. While face to face gesturing has natural cues to help identify who is gesturing, cursors do not. We label each cursor with a user's name. In addition (and more subtly) each cursor is oriented relative to the person's caricature ie all cursors are rotated $90^{o}$ from each other (see below and Figure 1).


- Cursor movements appear with no apparent delay on all displays, which means that they remain synchronized with verbal communication.
- Cursors always maintain their same relative location on every display so that they retain their relation to the work surface objects.

*Criterion 2: A collaborative tool should minimize the overhead encountered when storing information.* In a design session where one team member pauses to record information, the rest of the group may either wait, occupy the pause with individual work, or keep on going (leaving the scribe behind). In *GroupSketch*, any person may store a snapshot of the current work surface into their own private directories at any time. While that particular workstation display will 'freeze' for a short period, other workstations remain unaffected. Any person may restore their private images back to the public work surface at will.

*Criterion 3: A collaborative tool should convey the process of creating artifacts to express ideas.* To convey the process of creating artifacts, the content and quality of the shared information must be as rich as possible. In *GroupSketch*, any work surface action, no matter how small, is visible with no apparent delay on all participants' screens. Every movement of the cursor, every pixel that is drawn, and every letter typed is immediately broadcast to other screens and are therefore immediately visible. As with cursors, artifact creation and manipulation remains synchronized with accompanying speech.

*Criterion 4: A collaborative tool should allow intermixing of work surface actions and functions.* While most graphical interfaces have distinct modes for text entry and for drawing, *GroupSketch* has a nearly modeless interface. When no mouse buttons are depressed, the cursor is in the pointing gestural state. Drawing occurs as long as the left button is depressed, which also turns the cursor into a pen (Figure 1, Sam's action). Typing immediately inserts text at the current cursor location (Figure 2a). The cursor image changes to the pen, and automatically reverts back to the hand-shaped cursor after a reasonable pause in typing is detected (Figure 2b). One can even enter text and draw simultaneously by holding one hand on the keyboard and the other on the mouse (Figure 2c)!

*Criterion 5: A collaborative tool should enable all participants to share a common view of the work surface while providing simultaneous access and a sense of close proximity to it.* Perhaps the most difficult (and to our minds unresolved) design issue dealt with the tradeoff inherent when orientations differ. We chose to follow the WYSIWIS (what you see is what I see) approach (Stefik et al. 1987a), where everyone sees the same information and actions on the work surface with the same upright orientation. A weak concession is made to orientation by the caricatures situated around the work surface representing people seated around a table (Figure 1). As mentioned previously, cursors are also rotated with respect to the caricature in an attempt to provide a relation between the cursor and its owner. (A beneficial side effect of rotated cursors is that two or more cursors can point, draw, or erase very close to each other with minimal overlap.) Private drawing areas are not supported.

The WYSIWIS display and the presence of all participants' cursors promotes a close sense of proximity. As participants track other cursors, they naturally associate actions in the work surface with people who are executing those actions. In addition, simultaneity is fully supported. All participants have free and equal access to the work surface, with no technical hindrance to simultaneous activity.

*Criterion 6: A collaborative tool should facilitate the participants' ability to coordinate their collaboration.* As *GroupSketch* does not enforce any style of social protocol and as all participants are in direct control of their actions, the group is free to use whatever coordination method suits them (an argument favouring this approach is presented by (Dykstra and Carasik 1991).

**3.3 Architecture of *GroupSketch*.** GroupSketch is implemented on Sun workstations running Unix, connected by Ethernet. The design contains two features unusual in traditional interface design. It is a distributed groupware program supporting "instantaneous" shared views of a display, and it supports multiple cursors. The internal architecture of *GroupSketch* is briefly described.

Two architectural alternatives to constructing distributed groupware are the *centralized* and *replicated* approach (Ahuja, Ensor and Lucco 1990; Lauwers, Joseph, Lantz et al. 1990; Lauwers and Lantz 1990). In the centralized approach, a *conference agent* is interposed between one application and all users' workstations. The principle function of this agent is to multiplex output streams from the application to each participant's workstation, and to multiplex all participant's input into a single stream directed at the application. The primary advantage is that synchronization is easy, as it is all handled in one place. The disadvantage is that network traffic is heavier as both input and output must be broadcast to every workstation. In addition, the single conference agent and application could be a bottleneck as all activity must be channelled through it.

In the replicated approach, the application and conference agent are replicated on every machine. Each conference agent accepts input from other workstations, passes the input to its resident application, which then recomputes and generates the necessary output. The advantage is that network traffic is reduced, since only participant input is broadcast over the network. As long as the applications remain synchronized, the output generated by each application in response to the input should be the same.

*GroupSketch* follows the replicated approach, with the conference agent and application represented as a single process on each workstation. Processes communicate via Unix stream sockets. An additional *registrar* daemon process performs registration functions. The following example indicates how the registrar incorporates new participants into a *GroupSketch* session.

1 An incoming participant or late-comer connects to the registrar, opens its own communication port for other connections, and sends the port address along with the participant's name and caricature data to the registrar.
2 The registrar acknowledges the newcomer and informs other participants' *GroupSketch* process of the newcomer's address in the network.
3 Each *GroupSketch* process connects to the newcomer, with the nearest sending it the current state of the work surface image. The registrar is now out of the loop.

Processes communicate events to each other through only eight primitive events (Table 2).

| Event | Information passed |
|---|---|
| Registering a new user | host name, port number, name of participant, caricature |
| Unregistering a user | Id of participant |
| Moving cursors | Id of participant, cursor shape, new coordinates |
| Drawing a line | Id of participant, start and end coordinates |
| Erasing a region | Id of participant, coordinates of region |
| Listing | Id of participant, string location, string, cursor shape, location of cursor |
| Clearing screen | — |
| Image transfer | binary data of the work surface image |

**Table 2.** Communication protocol between processes

Multiple cursors presented a significant problem. Current window systems are inadequate as they only support single cursors, often implemented at a very low level or in the system kernel due to performance demands. Many systems also fix a maximum size to the cursor, a size we consider too small for our purposes. In *GroupSketch* we eschewed window systems completely in favour of a graphical library that allowed us to manipulate the bit-map display directly (we used the Sun Pixrect library). Multiple cursors are implemented directly by exclusively OR'ing bitmaps. Reading from the mouse device driver and writing to the screen provided efficient and fast responses. However, we are not certain if this approach is the best one. While the system performs well, designing a graphical interface from the ground-up is a major amount of work. For example, while menus are provided as high-level constructs in window systems, we had to build the menu look and feel from scratch

## 4. Observations of use.

*GroupSketch* was tested under relatively informal conditions. Six short (one hour) design meetings with two to four participants were held. In some cases, participants were in the same room but separated by two-meter high partitions, while in others they were in separate offices connected by telephone. Participants were given the choice of designing an advanced telephone or choosing a problem or task relevant to them. We observed the group's behaviour during the meeting, and collected subject's comments afterwards. We also observed *GroupSketch* use during an open house, where people of varying computer sophistication (from none to high) tried *GroupSketch*. The following observations are tempered by our own subjective interpretation of events.

*GroupSketch is very easy to learn.* People with even limited computer experience were able to use *GroupSketch* in less than a minute. We attribute this to the system's direct analogy to the paper sketchpad, the modeless nature of the interaction, and the simple syntax. Learning was at its best when a knowledgeable remotely-located *GroupSketch* participant taught a newcomer by providing examples of actions through the work surface and then watching the newcomer attempt those actions.

*GroupSketch is effective.* In spite of its simplicity, *GroupSketch* worked. Participants were able to pursue their tasks effectively, using strategies analogous to those observed in face to face design meetings.

*The worst part of GroupSketch is trying to draw with a mouse.* People expressed frustration when drawing with a mouse. A stylus would have been a large improvement.

*Increasing the number of participants in an open floor policy increases parallel activity but also decreases focused attention.* We observed much simultaneous activity. As noted by Tang, this comes at the price of reduced group attention (Tang 1989). For example, when four participants were collaborating, one person commented that she found it difficult to listen to another participant when others were actively writing or drawing in the communal work area. We expect this problem to be exacerbated as group size increases. Yet most participants agreed that restricting access to the work surface or introducing turn-taking would be unacceptable.

*Movement of the cursor synchronized with a participant's voice provides the greatest sense of tele-presence. The static caricature added nothing.* The presence of even idle cursors in the work surface was considered important by participants. People did not have problems distinguishing who was doing what. Still, the quality of presence did not match that of a face to face meeting. For example, we observed two occasions when visually separated but co-located participants involved in an intense discussion left their computers to speak face to face.

*The shared work surface captured participants' attention and focused interaction.* There is a strong focus of attention on the work surface. Participants' eyes remained fixed on the shared area for long periods of time, as if they did not want to miss any of the actions occurring in the work surface. The ease of drawing and talking simultaneously around artifacts seemed to provide a focused interaction.

*Participants desired greater functionality.* People familiar to computer systems wanted functionality greater than a simple sketchpad could provide. These included object-oriented drawing tools over free-hand bit-mapped sketching, editable text fields, and other features commonly available in single-user graphical packages.

*Intermixing listing and drawing (text and graphics) occurred frequently and naturally.* Resulting artifacts contained a good mixture of graphics and textual lists of points.

*Vertical orientation of the work surface removed the physical limitations of the table top.* Users had no problem recognizing objects on the display. As people could literally draw on top of one another, we observed people working together on objects in quite close proximity (examples include multiple people erasing different parts of a single line and cooperative construction of a drawing artifact).

*The work surface is too small.* The work surface quickly becomes cluttered during long design sessions, especially with larger group sizes. Larger displays, windowing strategies, or better storage and retrieval facilities are required.

## 5 Perspective.

People using *GroupSketch* are struck with the feeling that its design just follows the rules of common sense. After all, it appears self-evident that multiple cursors should be available, that simultaneous activities should be allowed, that all aspects of the drawing process should be visible, and so on. Is it really necessary to go to all the bother of studying group behaviour, of deriving design guidelines, and of evaluating the resulting system?

The best answer to this question is to show examples of related groupware systems that have failed to live up to these seemingly self-evident criteria. Early collaborative systems, for example, combined voice conferencing with facsimile transmission, use of electronic boards, or slow scan video. While all participants could see the same resulting information, gestures were not visible nor could one see the fine-grained process of creating artifacts. *View-sharing systems* that allow people to share views and interactions with single-user applications demand serial rather than sequential interaction for technical reasons, and multiple cursors for gesturing are rarely supported (Greenberg 1990).

Several systems designed explicitly as a group sketchpad have also failed to live up to "the rules of common sense". Consider Xerox PARC's *Boardnoter*, a computerized whiteboard used to support face to face meetings (Stefik et al. 1987a; Stefik et al. 1987b). While a single large tele-pointer could be seen by all, individual cursors were not. Neither did participants see each others actions as they occurred, for actions were not broadcast until a complete graphical stroke was made or a complete text line entered. *Xsketch*, a recent object-based group drawing package suffers a similar lack as its objects are only transmitted after they are created (Lee 1990). *WScrawl*, a group sketchpad in the public domain, does not show multiple cursors.

On the positive side, there are several systems (including *GroupSketch*) that do support the kinds of interactions people expect from a group sketchpad. All have one thing in common: they were derived from Tang's design principles (Table 1). While these systems are quite diverse, they all share a common feel, and observations of use are strikingly similar. Two systems, for example, are video based. *VideoDraw* uses polarizing filters to fuse two video images together (Tang and Minneman 1990). Participants draw directly onto the video screen, and can see the other person's hand and drawing underneath. *TeamWorkStation*, on the other hand, uses hardware to fuse video signals (Ishii 1990). The advantage is that people can perform their activity on any work surface (such as a desktop), with a video camera recording and fusing its image with other work surface images. Both systems are limited by scalability, for serious image deterioration results when too many images are fused.

*Commune* is a workstation-based system built independently but in parallel with *GroupSketch* (Bly and Minneman 1990; Minneman and Bly 1990; Minneman and Bly 1991). The interface to the two systems are remarkably similar, with a few minor exceptions. In *Commune*, the monitor is oriented horizontally. People write directly on top of the monitor with a stylus—the resulting artifacts are superior to the ones generated on our mouse-based system. (An interesting side issue is whether *Commune* users will ignore the areas of the screen occluded by their arms.) *Commune* architecture is hard-wired together, and currently supports a maximum of three collaborators (*GroupSketch's* limitation to four participants is easily removed). In spite of these differences, Bly's observations of *Commune* use are in accord with ours (Bly and Minneman 1990).

*CaveSketch* is another very recent workstation-based drawing tool heavily based upon Tang's design criteria (Lu and Mantei 1991). Unlike the above systems which support a single work surface (analogous to a single sheet of paper), *CaveSketch* provides multiple drawing layers that can be superimposed on one another. When used effectively, layers can provide a sense of ownership (individuals can draw out ideas on their own layer), awareness of other participant activities (activities on other layers are visible in a lighter color), information hiding and revealing (subsets of layers can be displayed), and so on (the set of group activities supported by the layering feature is listed in Lu and Mantei 1991).

*GroupSketch* and related systems attempt to support small group design by providing a common work surface, a group sketchpad. We consider these systems usable not because of any underlying technical wizardry, but because they were derived from observations of group interaction with a shared work surface in face to face meetings. In the near future, we expect much more functionality in group tools, and that these tools will allow the group to accomplish far more than what it could normally do with paper and pencil.

***Acknowledgement.*** The Alberta Research Council played a significant role in the development of *GroupSketch*. Ralph Bohnet gained early experiences in groupware design on a separate project while interning there; Saul Greenberg was given release time to supervise the project as an Adjunct Professor at the University of Calgary; and a summer intern was provided to refine the final stages of the original buggy software prototype.

GroupSketch is available at no cost via anonymous ftp from the University of Calgary, Department of Computer Science. Contact Saul Greenberg for further information.

## References.

Ahuja, S. R., Ensor, J. R. and Lucco, S. E. (1990) "A comparison of applications sharing mechanisms in real-time desktop conferencing systems." In *Proceedings of the Conference on Office Information Systems*, p238-248, Boston, April 25-27.

Bly, S. (1988) "A use of drawing surfaces in different collaborative settings." In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88)*, p250-256, Portland, Oregon, September 26-28, ACM Press.

Bly, S. A. and Minneman, S. L. (1990) "Commune: A shared drawing surface." In *Proceedings of the Conference on Office Information Systems*, p184-192, Boston, April 25-27.

Dykstra, E. A. and Carasik, R. P. (1991) "Structure and support in cooperative environments: The Amsterdam Conversation Environment." *Int J Man Machine Studies*, **34**(3), March. In the special edition on CSCW & Groupware.

Egido, C. (1988) "Video conferencing as a technology to support group work: A review of its failures." In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88)*, p13-24, Portland, Oregon, September 26-28, ACM Press.

Galegher, J., Kraut, R. and Egido, C. (ed.) (1990) *Intellectual teamwork: Social and technological foundations of cooperative work*. Hillsdale, NJ, Lawrence Erlbaum Associates.

Greenberg, S. (1990) "Sharing views and interactions with single-user applications." In *Proceedings of the ACM/IEEE Conference on Office Information Systems*, p227-237, Cambridge, Massachusets, April 25-27.

Greenberg, S. (1991a) "Computer supported cooperative work and groupware: An introduction to the special edition." *Int J Man Machine Studies*, **34**(2), February. In the special edition on CSCW & Groupware.

Greenberg, S. (1991b) "An annotated bibliography of computer supported cooperative work." *ACM SIGCHI Bulletin*, July.

Greenberg, S. and Chang, E. (1989) "Computer support for real time collaborative work." In *Proceedings of the Conference on Numerical Mathematics and Computing*, Winnipeg, Manitoba, September 28-30. Available in Congressus Numerantium vol 74 and 75.

Greif, I. (1988) *Computer-supported cooperative work: A book of readings.* Morgan Kaufmann Publishers Inc, San Mateo, California.

Grudin, J. (1989) "Why groupware applications fail: Problems in design and evaluation." *Office: Technology and People*, **4**(3), p245-264.

Ishii, H. (1990) "TeamWorkStation: Towards a seamless shared space." In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90)*, p13-26, Los Angeles, California, October 7-10, ACM Press.

Johansen, R. (1988) *Groupware: Computer Support for Business Teams.* The Free Press, Macmillan Inc., New York.

Johansen, R. and Bullen, C. (1984) "Thinking ahead: What to expect from teleconferencing." *Harvard Business Review*, p4-10, March/April. Reprinted in Greif, 1988.

Lauwers, J. C., Joseph, T. A., Lantz, K. A. and Romanow, A. L. (1990) "Replicated architectures for shared window systems: A critique." In *Proceedings of the Conference on Office Information Systems*, p249-260, Boston, April 25-27.

Lauwers, J. C. and Lantz, K. A. (1990) "Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems." In *Proceedings of the ACM/SIGCHI Conference on Human factors in Computing*, Seattle Washington, April 1-5, ACM Press.

Lee, J. J. (1990) "Xsketch: A multi-user sketching tool for X11." In *Proceedings of the Conference on Office Information Systems*, p169-173, Boston, April 25-27.

Lu, I. and Mantei, M. (1991) "Idea management in a shared drawing tool." Research report , Department of Computer Science, University of Toronto, Toronto.

Minneman, S. L. and Bly, S. A. (1990) "Experiences in the development of a multi-user drawing tool." In *The 3rd Guelph Syposium on Computer Mediated Communication*, p154-167, Guelph, Ontario, Canada, May 15-17, University of Guelph Continuing Education Division.

Minneman, S. L. and Bly, S. A. (1991) "Managing a trois: A study of a multi-user drawing tool in distributed design work." In *ACM SIGCHI Conference on Human Factors in Computing Systems*, New Orleans, April 28-May 2, ACM Press.

MIT (1983) "Talking heads." In *Discursions*, Boston, Mass, Architecture Machine Group, MIT. Optical disc.

Stefik, M., Bobrow, D. G., Foster, G., Lanning, S. and Tatar, D. (1987a) "WYSIWIS revised: Early experiences with multiuser interfaces." *ACM Trans Office Information Systems*, **5**(2), p147-167, April. An earlier version appeared in CSCW '86.

Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S. and Suchman, L. (1987b) "Beyond the chalkboard: Computer support for collaboration and problem solving in meetings." *Comm ACM*, **30**(1), p32-47. Reprinted in Greif, 1988.

Tang, J. C. (1989) "Listing, drawing, and gesturing in design: A study of the use of shared workspaces by design teams." PhD thesis, Department of Mechanical Engineering, Stanford University, California, April. Also available as research report SSL-89-3, Xerox Palo Alto Research Center, Palo Alto, California.

Tang, J. C. (1991) "Findings from observational studies of collaborative work." *Int J Man Machine Studies,* **34**(2), February. In the special edition on CSCW & Groupware.

Tang, J. C. and Leifer, L. J. (1988) "A framework for understanding the workspace activity of design teams." In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88)*, p244-249, Portland, Oregon, September 26-28, ACM Press.

Tang, J. C. and Minneman, S. L. (1990) "Videodraw: A video interface for collaborative drawing." In *ACM SIGCHI Conference on Human Factors in Computing Systems*, p313-320, Seattle Washington, April 1-5, ACM Press.