

WYSIWIS Revised: Early Experiences with Multiuser Interfaces

M. STEFIK, D. G. BOBROW, G. FOSTER, S. LANNING, and D. TATAR
Xerox Palo Alto Research Center

WYSIWIS (What You See Is What I See) is a foundational abstraction for multiuser interfaces that expresses many of the characteristics of a chalkboard in face-to-face meetings. In its strictest interpretation, it means that everyone can also see the same written information and also see where anyone else is pointing. In our attempts to build software support for collaboration in meetings, we have discovered that WYSIWIS is crucial, yet too inflexible when strictly enforced. This paper is about the design issues and choices that arose in our first generation of meeting tools based on WYSIWIS. Several examples of multiuser interfaces that start from this abstraction are presented. These tools illustrate that there are inherent conflicts between the needs of a group and the needs of individuals, since user interfaces compete for the same display space and meeting time. To help minimize the effect of these conflicts, constraints were relaxed along four key dimensions of WYSIWIS: display space, time of display, subgroup population, and congruence of view. Meeting tools must be designed to support the changing needs of information sharing during process transitions, as subgroups are formed and dissolved, as individuals shift their focus of activity, and as the group shifts from multiple parallel activities to a single focused activity and back again.

Categories and Subject Descriptors: D.2.2 [Software Engineering]: Tools and Techniques—*user interfaces*; H.1.2 [Models and Principles]: User/Machine Systems—*human factors*; *human information processing*; H.4.2 [Information Systems Applications]: Types of Systems; H.4.m [Information Systems Applications]: Miscellaneous

General Terms: Design, Human Factors

Additional Key Words and Phrases: Collaborative systems, computer-supported collaboration, computer-supported group work, computer-supported meetings, multiuser interfaces, WYSIWIS

1. INTRODUCTION

The Colab is an experimental meeting room at Xerox PARC in which computers support collaborative processes in face-to-face meetings. The Colab is designed for small working groups of two to six people (see Figure 1) using personal computers connected over a local-area network.

The Colab project is supported in part by grant N00140-87-C-9258 from the Defense Advanced Research Projects Agency.

A previous version of this paper was published in the *Proceedings of the Conference on Computer-Supported Cooperative Work* (Dec. 3-5, 1986), pp. 276-290.

Authors' address: Intelligent Systems Laboratory, Xerox Palo Alto Research Center, Palo Alto, CA 94304; stefik.pa@xerox.com

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1987 ACM 0734-2047/87/0400-0147 \$01.50

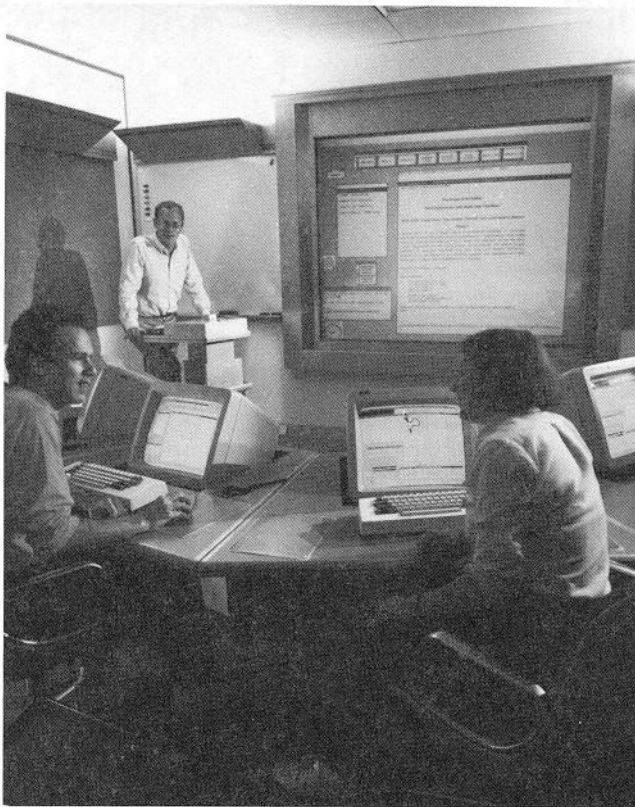


Fig. 1. View of the ISL Colab. The Colab is an experimental meeting room that provides computational support for collaboration in face-to-face meetings. It is designed for typical use by two to six persons. Each person has a workstation connected to a personal computer. The computers are linked together over a local area network (Ethernet) which supports a distributed database. In addition to the workstations, the room is equipped with a large touch-sensitive screen and a stand-up keyboard. (Photograph by Brian Tramontana).

Some simple observations prompted the formation of the Colab project. Office workers spend 30 to 70 percent of their time in meetings and office automation does not reach people who are away from their offices [8]. Although we have been developing computer tools and programming environments for several years, whenever we enter a conventional meeting room we must abandon them and fall back on using a chalkboard.

Computational meeting tools are being designed and used in the Colab for a range of informal and formal meeting processes. These tools, which are described elsewhere [5, 10], support various kinds of meetings, such as meetings for organizing ideas and meetings for developing and evaluating competing design proposals. *Multiuser interfaces* are essential to these tools. Multiuser interfaces provide meeting participants with simultaneous and shared access to the meeting database.

WYSIWIS (What You See Is What I See—pronounced “whizzy whiz”) is an abstraction that guides the design of multiuser interfaces. It is an abstraction of some of the functional properties of a chalkboard in a meeting. In “strict” WYSIWIS, everyone sees exactly the same image of the written meeting information and can see where anyone else is pointing. Colab tools support this abstraction by maintaining synchronized views across workstations and providing facilities for *telepointing* with publicly visible cursors.

Although strict WYSIWIS is an important abstraction for multiuser interfaces in the Colab, it is too inflexible, and all of the meeting tools need some relaxed interpretation. For example, many tools provide both *private* and *public* windows, that is, single-user and multiuser windows. By providing personal access to information, private windows mimic the concept of a personal notepad more than a public chalkboard. Strict WYSIWIS would outlaw private windows.

This paper is about the ways in which the WYSIWIS abstraction for multiuser interfaces needs to be relaxed in order to accommodate important interactions in meeting processes. There are four dimensions, with corresponding constraints in the strict WYSIWIS definition: space, time, population, and congruence. The following relaxations have proved useful. Strict WYSIWIS applies to everything on the displays; applying it to only a subset of the visible objects (e.g., windows and cursors) relaxes the space constraint. Strict WYSIWIS requires that images be synchronized; allowing delays in updating or viewing relaxes the simultaneity constraint. Strict WYSIWIS requires shared viewing to apply to everyone in the full meeting group; allowing sharing to be limited to subgroups relaxes the population constraint. Strict WYSIWIS requires that images be identical; allowing alternative views (including visual variations) relaxes the congruence constraint.

The presentation is organized around two case studies that deal with progressively more subtle interactions. We begin with a multiuser interface for freestyle sketching. This case raises issues that are preliminary to the main study of this paper: multiuser interfaces for a group collaborating on the organization of ideas. The design issues raised in this paper are the result of our reflections on our use of the first generation of collaborative meeting tools. They illustrate how the computer media must accommodate the needs of a group, as well as the needs of individuals.

2. CASE 1: THE HUMBLE CHALKBOARD

In the Colab we sometimes refer to the chalkboard¹ as the competition, that is, the competing technology for meetings. Chalkboards are quite effective for many things, such as flexible placement of text and figures, ease and familiarity of use, and providing a central focus for group attention. Chalkboards can be used to support a wide range of meeting processes. The WYSIWIS abstraction comes

¹ We use the term *chalkboard* to refer to any of the wall-mounted erasable writing surfaces, commonly used in meeting rooms, which are white, black, or some other color and upon which the marks are made with chalk, crayon, or ink. We chose this term to avoid misunderstandings about the word *blackboard*, which means, among other things, a commercially available teleconferencing product and a programming organization for artificial intelligence systems. We also avoid the term *whiteboard*, which means, among other things, a white metal writing surface on which colored felt-tipped pens are used, and a particular graphical database tool developed at Xerox PARC [4].

from our observation of the role of a chalkboard as a recording center and focus of group attention in meetings. Our primary assumption is that this abstraction is useful for multiuser interfaces.

Chalkboards also have some flaws. Space on a chalkboard is limited. Rearranging items is awkward and is done by copying items by hand and then erasing the originals. Information storage is unreliable because chalkboards are often erased between meetings. Information cannot be changed, except by the person at the chalkboard, or carried away by meeting participants, unless they copy it manually. Indeed, information on a chalkboard is isolated: It cannot easily be copied, transmitted, transformed, or accessed by computers.

Computer tools can maintain the strengths of the chalkboard while improving on some of its shortcomings. Text and figures in a computer medium can be made legible and easily movable. In window systems, one can quickly reorganize the information on display. Data can be stored reliably in databases between meetings. Computer programs can put information in reach of all meeting participants. Best of all, the computer medium is active and can enable many kinds of operations that are unthinkable with a chalkboard.

This is not to say that the issues about computer support of meetings are understood and solved. Many issues arise in finding ways to understand trade-offs between conflicting needs for use of limited display space. In this paper we try to articulate and separate the issues that arise in a multiuser context and to propose some solutions to fit available technology and our meeting processes.

Before starting the report of the case studies, we need to identify one more important assumption that underlies our work. It derives from widespread practice in single-user interfaces but takes on special importance in a multiuser setting. In many of our childhood and everyday situations, collaboration skills are acquired around the joint manipulation of physical objects. Examples include passing a ball in sports, passing a tool or a part in a collaborative assembly or repair task, and handling paper and staplers in a collaborative collating task with an old-fashioned copier.

The Colab meeting tools attempt to draw on this familiar kinesthetic and spatial experience of teamwork by representing information and operations in terms of visible, manipulable objects. Our designs of multiuser interfaces are based in part on widely known examples of single-user interfaces in which one can treat icons in a display as objects, and manipulate them by moving them around. This approach, sometimes called direct manipulation, draws on much previous work and contrasts with more limiting linear representations of information, such as streams of characters for text [9].

2.1 Freestyle Sketching

Of all the Colab meeting tools, *Boardnoter* is the one that most closely imitates the informal functionality of a chalkboard. Our goal has been to make Boardnoter as easy to use as a chalkboard. Hoping to capitalize on habits acquired in elementary school, we decided further that drawing with Boardnoter should be similar to drawing on a chalkboard. For example, to sketch a square with Boardnoter one can just “pick up the chalk” and make four strokes.

Figure 2 illustrates the general layout of this meeting tool. The main feature is a large writing area (the board). To draw on the board one uses the chalk; to

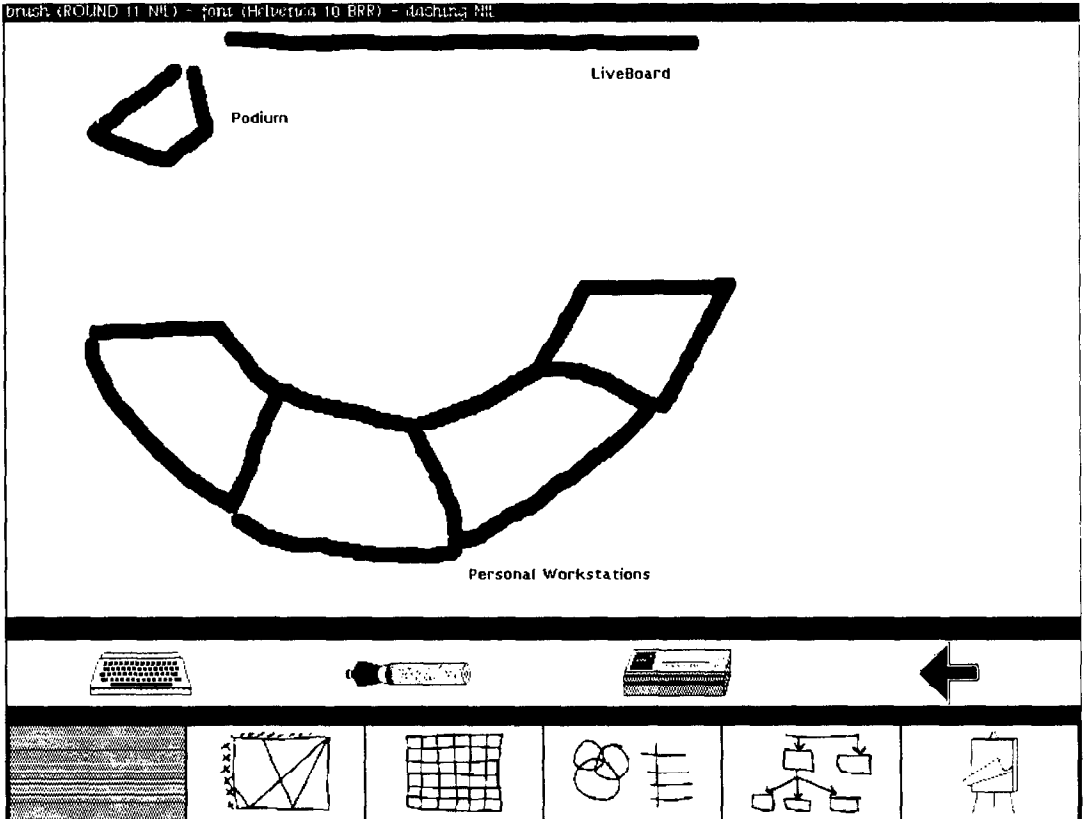


Fig. 2. Screen image of Boardnoter. The Boardnoter meeting tool in Colab is operational but still in its early stages of development. A key feature of the tool is that it can provide a large area for free-style sketching. Below the writing area is a "chalk tray" on which several implements appear: a piece of chalk, an eraser, a keyboard, and a pointer. To draw on the board, one uses the chalk (picked up by clicking the mouse or pen over the chalk icon); to erase one uses the eraser; to point one uses the pointer. More than one boardful of information may be needed in the course of a meeting. One can shrink the current board to a miniature version, save it in the "stampsheet" at the bottom, and replace it with a clean slate or one previously stored.

erase one uses the eraser; to type one uses the keyboard; to point one uses the pointer. Collectively we shall refer to these as *implements*; they are stored in the chalk tray below the board. Following the WYSIWIS abstraction, all of the actions with these implements that change the image on the board are replicated on the displays of the other meeting participants.

Interaction with Boardnoter can be through different kinds of physical pointing devices such as a mouse or a cordless pen. In either case, clicking on an implement icon causes the screen cursor to display that implement and makes Boardnoter ready to work as follows:

—To use the chalk, one clicks on the chalk icon, causing the cursor to appear as a piece of chalk. For familiarity at PARC, we actually depict the chalk as a whiteboard pen. However, to maintain the chalkboard terminology in this paper and to avoid ambiguity with the physical pointing device (preferably a cordless

pen), we shall consistently refer to the tool's writing implement as the *chalk*. Moving the pen moves the chalk cursor. If the pen button is depressed while being moved, a trace of the movement (the "chalk mark") is left on the screen. The shape and size of the chalk can be adjusted by separate controls.

—To use the eraser, one clicks on the eraser icon, causing the cursor to appear as an eraser. Moving the pen across the screen causes items near the eraser to fade—shifting from black to gray. If the pen is clicked, the gray items disappear. This action by the eraser is different from that of a chalkboard eraser: The computational equivalent of a chalkboard eraser would treat a board as a collection of pixels. However, each stroke of the Boardnoter chalk is remembered as a single item. When an eraser is near, an entire item is grayed out. This makes many erasing tasks easier. For example, it eliminates the need for small motions to erase particular pixels when two items are very close together.

—To use the keyboard, one clicks on its icon, causing the cursor to appear as a miniature keyboard. The cursor can then be moved. Clicking again causes the cursor to flash, and one can use the workstation keyboard to enter text that is then displayed at the cursor position.

—To use the pointer, one clicks on the pointer icon, causing the cursor to appear as a large pointing arrow. In contrast with the pen, moving the pointer around does not leave behind chalk marks. However, it does cause a pointer image to move around on the displays of the other meeting participants. We call this *telepointing*. ("Lookee here!") In addition, clicking the mouse while pointing causes an image of the pointer icon to be deposited on the board. This image is analogous to the small magnetic arrows that are sometimes used on whiteboards. Like chalk marks and typewritten text, these deposited pointers can later be erased.

The tangibility of these implements and displays marks is a simple example of the direct manipulation abstraction mentioned earlier. Implements are picked up, moved about, and used. The pictorial moding of the cursor draws on a natural moding of the chalkboard: We can have only one implement in our writing hand at a time, and generally we use only one hand.

2.2 WYSIWIS Considerations

The Colab provides separate workstations for each participant in a meeting. A workstation provides a kind of immediate access that a chalkboard across the room cannot. One can call up particular information or add or change information without leaving one's chair. One motivation for this is to enable parallel activity.

All of the Colab tools are based on WYSIWIS as a foundational abstraction for supporting parallel activity with multiuser interfaces. We have found WYSIWIS useful both as a basis for the initial designs of interfaces and as a guide to revising and changing them. We have discovered that WYSIWIS, although crucial, is too inflexible if strictly interpreted, and must be relaxed to better accommodate important interactions in meetings. Our thinking about meeting tools is focused by the four key dimensions of WYSIWIS: display space, time of display, subgroup population, and congruence of view.

In Boardnoter, parallel activity is enabled by giving participants their own implements. A strict interpretation of WYSIWIS would require that all of the implements and their movements be visible to all of the participants. However, we quickly discovered that the movement and flickering of multiple cursors became distracting when multiple participants were active.

Issue. The WYSIWIS display of cursors from multiple users is unacceptably distracting.

Design Solution. Display only local implements and those used for telepointing. This relaxes the WYSIWIS constraint on display space and recognizes the distinction between cursors being used to provide focus for a group and their use only for the pointing needs of individuals.

With our current technology, Colab participants coordinate by voice and the display of information. This brings us to our next WYSIWIS issue. At what grain size should changes to a display be updated? For example, when one participant is drawing with the chalk, should the other participants see every small movement being made? When one participant is typing text on the chalkboard, should other participants see every keystroke? The current version of Boardnoter shows only the drawing of a completed stroke and the typing of a completed text entry. As noted by Greif, choice of grain size can be important for system performance by greatly reducing the rate of changes to the database and the rate of updates to the displays [6]. But large grain-size updating also restricts interaction to a large grain size. For example, a participant must complete typing before others can see the text and cannot get help with phrasing while typing. However, after the text has been entered, it can be edited by any participant.

Issue. Small grain-size transmission of data allows small grain-size collaboration but is computationally expensive.

Design Solution. Broadcast changes to information when the user indicates completion or after a reasonable time interval. Although this argues for relaxing the WYSIWIS simultaneity constraint to achieve performance, there are other reasons in meeting situations for being able to control the granularity of interaction. For example, it may be useful to allow moments of privacy while a participant works out the spelling of a word or a turning of phrase.

Sometimes it is important to make certain suboperations visible to enable small grain-size collaboration. For example, in collaborative editing, we have found it useful to show publicly items selected for deletion or rearrangement before the operation is carried out.

Issue. Object selection is sometimes a natural breakpoint in operations for collaborative editing, such as in moving, copying, and deleting objects.

Design Solution. Provide transmission of teleselection on demand. This recognizes that there may be important “joints” in public processes that need to be visible in order to accommodate discussion.

Issue. Teleselections can be confusing if they are displayed in the same way as local selections. Confusion is especially likely in the middle of local operations that require their own selections.

Design solution. Modify the display of teleselected objects so that they are distinguishable from objects locally selected, for example, by using different shades of gray. This solution relaxes the WYSIWIS congruence constraint by reflecting the fact that multiple operations can be taking place at the same time.

One of the important characteristics of real chalkboards is that they are large enough to display a substantial amount of pictorial and written information. Their size is limited mostly by human reach. When we take into account the size that text characters need to be drawn to be visible in a standard-sized conference room, it turns out that the amount of information (measured in number of text characters) that can be usefully placed on a chalkboard requires a large portion of a high-resolution computer display.²

To bring more writing space within reach, classrooms are sometimes fitted with sliding chalkboards. Many conference rooms are fitted with multiple flip-charts so that a large amount of information can be scanned at once. Analogously, icons enable more effective use of display space by providing recognizable reminders in shrunken form [2, 3]. In addition, icons can be arranged in arrays for easy reference, much like paned windows [1]. We use the term *stampsheet* for such arrays of icons.

Issue. Meetings often need multiple chalkboards, but there is room to show only one on a computer display.

Design Solution. Boardnoter provides multiple visible boards by arranging shrunken versions of them as icons in a stampsheet below the chalk tray. (See Figure 2).

Clicking on a stamp causes the current large board to be redrawn in miniature scale in its own place in the stampsheet and causes the selected stamp-size board to be expanded and made available for use as a full-size chalkboard. Using the stamp marked "new board," one can obtain a fresh chalkboard. Since the stamps portray the contents of the boards in shrunken form, it is easy to recognize them and to return to a previous board. By itself, the stampsheet concept is not a relaxation of WYSIWIS because all participants continue to see the same information, shrunken or not. But as we will see later, stampsheets and their extensions lead to a family of WYSIWIS relaxations.

Issue: Pixel-based computations for shrinking boards are computationally expensive and do not always produce recognizable images.

Design Solution: Images are represented in terms of scaled projections of objects, rather than raw pixels, and provide specialized shrinking methods for

² To compare the amount of space on a computer display with a standard 4-by-5-foot chalkboard, we carried out a simple, informal measurement. We filled both media with text at the smallest size that was easily readable. We required that the text appear approximately the same size if viewed from a distance of 10 feet from the chalkboard or from a distance of 2 feet from the computer display. This corresponded roughly to characters in 12-point font on the computer displays and characters 15 millimeters in height on the chalkboard. The chalkboard could display approximately 3520 characters (32×110). The current Colab uses displays that are 808-by-1024 pixels, which were able to display 9520 characters (56×170). Even allowing for the space-displaying implements and the stampsheet, Boardnoter provides the same spaciousness as a chalkboard in less than half of the computer screen.

different kinds of graphics objects. This solution relaxes WYSIWIS congruence by allowing systematic distortions, in addition to changes of scale.

Our wish to have recognizable small versions of boards is another argument for maintaining graphics objects with specially tuned techniques for redrawing them at different scales (e.g., letters in smaller fonts). However, our experience with this is quite limited, and we are not sure that this technique will work with all images.

2.3 Beyond the Chalkboard

Our initial focus on Boardnoter was to imitate the capabilities of a real chalkboard. In the next phase of development we will add to Boardnoter capabilities beyond those of the conventional chalkboard. For the operations described so far, we want to preserve the analogy to the familiar chalkboard. The next version of Boardnoter will add capabilities for copying, moving, resizing, linking with rubber-band lines, grouping, and smoothing (neatening), and also for using and scaling selections from a set of predrawn images. Our experience with other Colab meeting tools suggests that several of these features raise questions related to WYSIWIS.

Although a meeting participant will usually focus on one board at a time, being able to see several of them simultaneously is important for both reference and copying. Imagine conferences at which chalkboards were kept in separate rooms so that participants had to run back and forth. That would get in the way of operations in which more than one board must be visible, such as when the contents of two boards are being compared or information is being copied across boards.

We have considered several different models for copying items between chalkboards. One model is the register model, which provides a constantly accessible place into which items can be moved intermediately during transit between boards. A variation of this general approach is the clipboard on the Macintosh.³ Following the direct manipulation abstraction, we prefer an alternative approach in which one can pick items up and move them around. This requires that multiple chalkboards be made accessible at the same time, for example, in a two-by-two arrangement of half-size boards. From this arrangement one can pick up copies of items from one chalkboard and deposit them in another.

Raising the possibility of multiple accessible chalkboards in Boardnoter highlights some assumptions and issues about its design so far. If one participant switches to a two-by-two format, should the displays of all the participants be switched? Having display space for four boards could allow different people to display different boards at the same time. Should WYSIWIS congruence and subgroup population constraints be relaxed to allow participants to view different boards?

We cannot answer these questions from our limited experience with Boardnoter and its associated formal and informal meeting processes. Although Boardnoter is operational, we need to extend its capabilities in order to carry out the necessary

³ Macintosh is a trademark of McIntosh Labs., Inc.

experiments. Nevertheless, a very similar set of questions arises in the redesign of a Colab tool with which we now have several months of experience: our Cognoter meeting tool for organizing ideas. That experience and interface redesign is discussed next.

3. CASE 2: COLLABORATION IN THE ORGANIZATION OF IDEAS

Cognoter, a meeting tool for organizing ideas for a presentation, supports a meeting process in which participants come together, usually without having prepared any materials. They determine the audience and goals for their presentation, the topics to be included, and the overall organization. The output of Cognoter is an annotated outline. Cognoter has been used for preparing many presentations and papers, including this one.

Since Cognoter is described elsewhere in detail [5, 10], the discussion here will be brief. We focus on the breakdown of WYSIWIS in the first design, and trade-offs in the redesign.

In organizing ideas for a presentation, several things can reduce the quality of the results. The presentation could fail to include some important topics; it could dwell on irrelevant or unimportant topics, or it could address the topics in an incoherent order. Cognoter separates these concerns by organizing the available actions into stages.

Cognoter is more formal than Boardnoter in both its representations of information and its meeting process. It organizes its meeting process into a sequence of stages: brainstorming, ordering and grouping, evaluation, and outline generation. These stages are additive in several senses. The database of decisions becomes more complete as a group advances through the stages. Although each stage is intended for completion of a particular part of the task, it is always possible to use the operations of previous stages to wrap up loose ends. One need not decide that some part of the process is absolutely completed before moving on. In addition, the operations in Cognoter are small and incremental.

3.1 The Stages of Cognoter

Each stage in a Cognoter meeting focuses attention on one part of the task of organizing ideas for a presentation. We originally adopted this structure because a similar one has proved useful in noncomputational settings; however, there are significant differences in the uses and effects of the stages when computers are introduced.

3.1.1 *Brainstorming Ideas.* The brainstorming stage is designed to foster the free-flowing contribution of ideas. There is one basic operation: A participant selects an empty space in a public window and types in a catch word or phrase characterizing an idea.

Unlike brainstorming in meetings without computers, there is no waiting for turns in Cognoter; any participant can enter an item at any time. Often the inspiration for an idea is triggered by another participant's actions—either saying something or entering an item in a public window. All items appear on everyone's screens. In addition, participants can augment an item with a longer description to help clarify its meaning.

3.1.2 *Organizing Ideas.* The order of ideas for the presentation is established in Cognoter by incremental and local steps. There are two operations: linking items into presentation order, and putting items into delineated subgroups. If item *A* is linked to *B* (meaning *A* comes before *B*), and *B* is linked to *C*, then *A* comes before *C*. If item *A* is linked to a subgroup, then it comes before every item in the group. By means of these transitive and distributive operations, a small number of explicit links can tightly constrain the total order of items.

The linking operation often takes place in conjunction with an oral justification. For example, if “expenses” and “bottom line results” were items, a participant might argue “We have to talk about expenses before bottom line results because otherwise the reader will not understand the results.” This relation is represented visually in Cognoter as an arrow linking the item labeled “expenses” to the one labeled “bottom line results.” It is also possible to move related ideas to a separate window. Before moving items, it is common practice to put them in a spatially compact cluster. This allows comment on the coherence of the proposed grouping.

In Cognoter, organizing becomes a richer task than in the traditional brainstorming paradigm. Not only do participants put ideas in a particular order, but they also tend to form subgroups that focus on the development of particular aspects of the subject matter. Since subgroups of ideas are usually put into separate windows, each subgroup of people tends to be focused around one window or another. We characterize these subgroups as *lightweight collaborations* because they are quickly formed and usually last only a few minutes.

Although there is parallel activity, there is also communication between subgroups. For example, one group may decide that some of the ideas in its windows do not fit with the others after all and may put them back into the general pool or offer to give them to another subgroup. Further assignment of responsibility for organizing sets of ideas requires additional communication. Finally, some activities require the attention of the whole group. When subgroups rejoin, participants recap the changes made in the subgroups.

3.1.3 *Evaluation of Ideas.* During the evaluation stage of Cognoter, the subgroup boundaries tend to dissolve, and the meeting participants function again as a single group. Participants try to understand the organization of the presentation as a whole. During this phase, ideas that seem irrelevant or less important than others can be deleted. Tentative outlines are generated by Cognoter upon request, and ambiguities in the ordering of ideas can be highlighted. Arguments are made about whether particular ideas are irrelevant or unimportant when compared with other ideas.

3.2 Some Problems with the First Cognoter

The use of the WYSIWIS abstraction in the first version of Cognoter was conceived in terms of the contents of individual windows and ignored issues of window placement. This approach drew on our experience with systems in which window shape, size, and position are completely under user control. We gradually came to appreciate an inherent conflict between our style of screen management and the needs of multiuser interfaces. Failing to address overall display layout

issues explicitly created the following unpredictable results for routine window manipulation:

—In Cognoter sessions, the displays of the participants often became crowded and cluttered with windows. In attempting to deal with this clutter, participants could close individual windows on their own displays. However, these windows reappeared whenever another participant made a change in them.

—When a participant changed the shape or position of a public window in Cognoter (e.g., to make more room for items), they were changed on all of the displays. This sometimes obscured private work on other's screens.

—The telepointer in Cognoter worked in terms of absolute screen coordinates but was used to point to items positioned relative to particular windows. Depending on the operations that have been done, a remote pointer could point to the wrong part of a display.

—For all these reasons, participants needed to fiddle with window sizes and placements too much. The problem was exacerbated by the fact that participants could have a mix of public and private windows. The private windows could be in different places on different displays, and participants were unaware of the different window placements. This caused unexpected occlusion.

The apparent bugs in Cognoter's original user interface resulted in part from a lack of principles for understanding inherent conflicts between the needs of an individual and the needs of a group. In a way, these conflicts reflect the combined use of the display for both private and public media—conventionally kept separately as a notepad and a chalkboard.

The following sections present a more detailed analysis of the differing needs of a group and its individuals in a face-to-face meeting. They suggest how the computer media can better accommodate the needs of both by relaxing the WYSIWIS constraints. The conflicts arise in part because the elements (e.g., windows) needed to support any of the activities must compete for the same display space and meeting time. In the following we compare two proposals for organizing the displays and use them in thought experiments to expose design principles for our second generation of meeting tools. The examples illustrate how the dimensions of WYSIWIS can be used to provide a guide for such thinking.

3.3 A WYSIWIS Stampsheet Approach

Even with single-user applications, the space on a workstation display is limited and often crowded with windows. In WYSIWIS situations, contention for display space is even more problematic because the same area is shared by multiple participants. In Boardnoter, stampsheets are used to create more space for those windows that are actively being used. Although stampsheets relax WYSIWIS by reducing scale, they provide a simple means for locating information since it is generally still recognizable when displayed in miniature.

Issue. The screen can be crowded with windows used mainly by other participants.

Design Solution. Allow participants to select independently which windows are at full scale. This relaxes WYSIWIS space and congruence constraints and

allows participants to reclaim display space for private use as they withdraw from parts of the public discussion.

One of the important properties of the Cognoter meeting process is the tendency to form lightweight collaborations of one to three people. Unfortunately, when active windows are reduced to static shrunken icons, participants lose the ability to follow the action. They can no longer quickly determine which information is being worked on.

Issue. When windows are shrunk to the miniature scale of stamps, it is no longer possible to assess quickly the locus of activity or which information has changed since the windows were last viewed at full scale.

Design Solution. Unlike conventional icons, stampsheet icons should *actively* indicate when information is changing. Stamps can be augmented with specialized indicators that show different measures of change. Making the stamps less similar to the full-size windows relaxes the congruence constraint beyond miniaturization. Adding the activity indicators compensates in part for the diminished WYSIWIS of stampsheets and provides visual clues to the participant both to monitor meeting progress and to guide a transition back to full WYSIWIS participation in the meeting.

Accordingly, stamps can be made to flicker when changes are made in the corresponding full-scale windows. This makes it easier to see at a glance the locus of activity at any given moment. To minimize the distraction of flickering, shades of gray in limited regions can be used, rather than blinking or inverting the entire stamp.

To make it easier to determine how much information has changed, one can attach a bar along an edge which is initially white and becomes progressively darker as changes accumulate. Alternatively, the bar could be in the format of a thermometer that fills as changes accumulate.

Issue. Identifying recent changes is also important for full-size windows, when participants refocus on a window.

Design Solution. Provide a facility for highlighting recent changes. This relaxes the WYSIWIS congruence constraint because the indications show only on the requestor's display. This relaxation of WYSIWIS makes it possible to superimpose privately needed information on top of public windows, without distracting other members of the group.

Notification of change supports parallel activity by enabling individuals and subgroups to catch up with work done by others. Shared windows support parallel activity by enabling subgroups to share common information. By limiting sharing to a subset of windows, subgroups can focus on particular shared information and filter out that which is distracting. Thus notification and window sharing are two sides of support for subgroups.

Issue. Subgroups need to cause other group members to attend to a particular item without interfering with the activities of other subgroups.

Design Solution. Provide a way of using the telepointer to show only within subgroup boundaries. This solution relaxes the WYSIWIS population constraint

by recognizing that the demarcation of public versus private or group versus individual is too inflexible. Meetings can operate productively with subgroups of different sizes. This is an example of where the flexibility of the computer-based tools enables styles of activity that are less convenient when there is a single shared medium. It is analogous to the use of separate flipcharts by subgroups in a large meeting.

This raises the issue of membership in subgroups. One possibility is to have explicit actions for defining and modifying subgroups. A problem with this is that it presupposes notification that would not already be part of a meeting. One does not normally need to make a note on a blackboard in a small meeting to announce that a subgroup is being formed. Explicit notification places a continuing burden on the participants to keep this information up-to-date.

The WYSIWIS stampsheet proposal does not support subgroups as a *formally selected* set of people. Instead, communication is based on the interests of the participants. All viewers of a window see the same information. Subgroups are those who have a common focus. People can play bridging roles between subgroups or monitor activities in several subgroups by maintaining views of windows being used by several groups.

Subgroups can also encounter situations in which they want to momentarily get the attention of the larger group, even though the windows containing the information may not be visible on everyone's screen.

Issue. Subgroups need to be able to bring information to the attention of a full meeting, even though the windows containing the information may not be visible on everyone's screen.

Design Solution. Provide both subgroup and full-group telepointing. A full-group pointer signals a *remote window fault* in the corresponding stamps, causing it to be expanded for any participant who does not already have the window at full scale. Having two kinds of telepointing makes it possible to relax the congruence and population constraints of WYSIWIS in a controlled way, reflecting the varying needs of the group. Here the computer technology simplifies the transition to full WYSIWIS participation in a way that shared flipcharts cannot.

Issue. Sometimes a participant wants to join a subgroup but cannot find the stamp corresponding to a subgroup window. Asking a member of the subgroup to identify the window using a full-group telepointer would be disruptive for other members of the full group.

Design Solution. Two ideas help fix this. One is to provide labels for windows and their corresponding stamps so that participants can refer to them by name both verbally and visually. Another idea is to extend the subgroup telepointer to flash an indicator on remote stamps. The latter helps to support a transition in a meeting process during which WYSIWIS becomes more strictly enforced as a group becomes more cohesive.

We have now distinguished two kinds of telepointing. Subgroup telepointing displays the remote cursor in the windows of all other participants who already have the window at full scale and flickers a remote pointing indicator in the stamps of participants not viewing full-scale windows. Full-group telepointing

causes the corresponding stamps of all participants to expand and show the telepointer. The distinction reflects the differing needs for coordination in subgroups of different sizes.

One of the important operations for organizing ideas in Cognoter is clustering. As discussed earlier, this operation is typically begun by first moving the items near each other in a public window and calling for discussion. The items can then be formed into an explicit subgroup that is displayed in its own window. When a subgroup of ideas is created and placed in a new window, it is necessary to decide which participants will see the new window on their displays. If it is shown only on the display of the creating participant, then all other members of the subgroup must individually find the corresponding stamp and expand it. If the window is displayed on the screens of all the meeting participants, it will appear unexpectedly and may disrupt the activities of other subgroups.

Issue. When a new public window is created to support joint action on new information, putting the window on all of the displays may disrupt the activities of other subgroups.

Design Solution. New windows are spawned from existing ones. They are automatically placed on the displays of those participants for whom the original window is at full scale. For participants having the original window at stamp scale, the windows are added to the stampsheet. Limiting the sharing of information to a subgroup relaxes the population dimension of WYSIWIS and reflects the fact that discussion and the formation of subgroups tend to be continuous processes. Strict WYSIWIS can be relaxed to support this by displaying the new window on the screens of participants known to be interacting actively.

Returning to the issue of screen management, when a participant places a public window, it is done without knowledge of the placement of any private windows on the other screens. This leads to the possibility of unexpected window occlusions for other participants.

Issue. Although stampsheets reduce the contention, public and private windows still compete for display space. When a participant adds a new public window to a subgroup, it can occlude windows of other participants.

Design Solution. Participants can control the placement of all windows on their displays. New public windows may be repositioned after they appear. By convention, certain regions are used primarily for the placement of private windows. Allowing personalized window placement relaxes the similarity dimension of WYSIWIS. Another idea would be to provide a density map of screen usage that would guide a participant to place the new window in a place where it might provide the least occlusion. Again, the density map would relax WYSIWIS by balancing the needs of a subgroup—showing an individual where private windows tend to be located without revealing their contents.

3.4 A WYSIWIS Rooms Approach

Windows on the display of a participant can arise from several sources. They can result from the direct action of the participant, from the actions of other

members of a subgroup, and from remote telepointing from a separate subgroup. Displays can become cluttered with windows. In this regard, it is interesting to compare the WYSIWIS stampsheet mechanisms proposed so far to the Rooms mechanisms, suitably extended for multiuser situations [7].

The Rooms concept was designed to help with a problem that already arises in single-user applications: Displays provide quite limited space for holding information, when compared with the amount of information that can be placed on a large table using sheets of paper. The analysis of this approach draws on an analogy to memory paging systems for computers in which most of the action during short working periods typically occurs in small subsets, called working sets, of the total pages.

Rooms organize collections of windows into related screenfuls of information. Each screenful is a room. To work well, the approach requires that each room be big enough to hold a working set of windows. Performance of the system can be determined in part by measuring the rate of *window faults*—occasions on which a user needs to access information not visible on the display. In general, a user is expected to organize rooms around tasks. For example, there can be a mail room with electronic mail windows, a writing room with text editors and related tools, a programming room, and so on. Windows needed for more than one task can appear in as many rooms as needed.

In the following we revisit the WYSIWIS relaxation issues that were introduced in our discussion of the stampsheet approach. We propose alternative solutions based on multiuser extensions of Rooms. The basic approach is to associate subgroups of participants with separate rooms. In Cognoter, different rooms would be formed around subtopic areas.

Issue. The screen can be crowded with windows used mainly by other participants.

Design Solution. Provide separate rooms for each subgroup. Rooms are connected by *doors*. New rooms and doors between them are created as needed. Rooms relax the space and congruence constraints by accommodating the need for subgroups and individuals to maintain their workspaces.

Issue. When subgroups are located in separate rooms, it is no longer possible to assess quickly where group activity is, or what information has changed since the rooms were last viewed at full scale.

Design Solution. Create an overview room from which one can watch the overall activities of the meeting. Following the miniaturization technique from stampsheets, the overview room can contain miniaturized active images of the rooms. Augment the room images with indicators that show how much change there has been and where the activity is from moment to moment. This solution relaxes the WYSIWIS congruence constraint by providing more graceful transitions as participants join parts of an ongoing discussion.

Issue. Identifying recent changes is also important when a participant reenters a room.

Design Solution. Provide a facility for highlighting recent changes in a room. This relaxation of the WYSIWIS congruence constraint makes it possible to overlay privately needed annotations on information in public windows.

Issue. Subgroups need to cause other group members to attend to a particular item without interfering with the activities of other subgroups.

Design Solution. Provide a subgroup telepointer that shows only within a room. This solution relaxes the WYSIWIS population constraint by recognizing temporary independence of multiple subgroups.

Issue. Subgroups need to be able to bring information to the attention of a full meeting, even though the windows containing the information may not be visible on everyone's screen.

Design Solution. Provide a full-group telepointer that *teleports* all participants of the meeting into the room of interest. A "backdoor" would be provided to take participants back to the meeting rooms they were using before the interruption. An alternative solution, which may be less disruptive, is to add the referenced window into the rooms of all of the participants. The window can be closed afterward. In either case the mechanisms provide a transition to full WYSIWIS for that window for participation by the entire group.

Issue. Although rooms reduce the contention, public and private windows still compete for display space. When a participant adds a new public window to a subgroup, it can occlude windows of other participants.

Design Solution. Establish conventions so that certain regions of rooms are used primarily for the placement of private windows or a room-limited density map. Both approaches recognize that there is a tension between the public and private use of display space.

3.5 Comparison of Approaches

The WYSIWIS stampsheet and Rooms proposals have been developed in response to shortcomings of the multiuser interfaces in the first generation of Cognoter. In the following comparisons we treat them as separate and competing proposals in order to articulate some of the issues and trade-offs. A reliable evaluation of the ideas requires further implementation and experimentation.

The WYSIWIS stampsheet and WYSIWIS Rooms proposals provide mechanisms to alleviate window crowding, resulting from the use of the same display space for multiple meeting participants and from the use of the same space for private and public purposes. Both proposals depend on patterns of window access. At any stage in a meeting, participants focus their attention on a relatively stable subset of the windows.

The two approaches differ in their assignment of responsibility for placing public windows. In WYSIWIS stampsheets, public window placements are privately determined; in WYSIWIS Rooms, public windows are in the same place for everyone. This reflects a difference in attitude: Rooms creates a more formal division of information, and it seems natural to expect a room to be set up when one enters. In both cases, telepointing is window relative.

Compared with stampsheets, the WYSIWIS Rooms approach creates harder boundaries between subgroups. Participants in a room are more isolated from the rest of the group because they have to leave the subgroup room in order to see the activities of other subgroups. To get an overview of activity, participants must go to the overview room.

There are some differences in the quality of activity overviews for stampsheets and Rooms. Since rooms keep subgroups more isolated, miniature images will be less recognizable to other subgroups that did not watch them evolve. Balancing this, the recent change indicators in both approaches may compensate by helping people to more gracefully join an ongoing discussion.

Sometimes, in the course of a Cognoter meeting, it is important to reconsider the division of items between two sets of ideas being worked on by two subgroups of people. For this purpose, a participant needs to make the relevant windows for the two subgroups simultaneously visible. In the WYSIWIS stampsheet approach, a participant can just open the relevant windows. In a WYSIWIS Rooms approach, there is a greater overhead because a participant must either create a new room for the comparison, or involve all the other members of one of the subgroups with the extra windows. With stampsheets, subgroups can more easily have ragged boundaries, with some participants playing bridging roles between subgroups.

In the WYSIWIS Rooms approach, rooms stay around even after participants finish working on a subgroup of ideas. In contrast, windows just return to their places in a stampsheet in the WYSIWIS stampsheet approach: no permanent record of subgroups is kept.

We conjecture that extra overhead will inhibit both the formation and dissolution of subgroups. Inhibiting the formation of subgroups will diminish the likelihood of lightweight collaborations. Inhibiting the dissolution of subgroups may encourage too much independence and dilute the sense of coherence for the group as a whole, leading to a greater difficulty in reintegrating ideas.

This section has focused on the differences between WYSIWIS Rooms and WYSIWIS stampsheets as independent approaches. Actually, the next generation of our tools may combine the ideas, such as providing rooms with stampsheets. Other fundamental assumptions of the proposals may also be reexamined. For example, both proposals have been written as if the only available displays were on the workstations of the participants. The Colab actually includes a large public display on one wall of the room as shown in Figure 1. This wall display could be used for public windows, which participants could also access from their workstations. Combining the wall display with WYSIWIS Rooms could enable a less jarring technique for full-group telepointing: Instead of teleporting participants to the appropriate room, the tools could simply direct their attention to the large public display.

Many of the design decisions for Colab meeting tools turn fundamentally on expectations for the size of the meeting group. The Cognoter tool was designed for a meeting process that involves 2 to 6 people. Meetings of this sort tend to last from 1 to 2 hours and yield outlines that require 2 to 5 pages of text. Since 20 stamps occupy less than 25 percent of the screen, ample space is still left for working windows. For larger groups or tasks, the approach may not work as well.

We hypothesize that WYSIWIS Rooms will prove increasingly effective with larger groups and larger tasks—just as larger physical rooms and breakout rooms are advocated in meetings not supported by computers.

4. SUMMARY

Our design of meeting tools begins with the assumption that multiuser interfaces should imitate the essential properties of a chalkboard as summarized by the WYSIWIS abstraction. Strict WYSIWIS means that everyone sees exactly the same image of the meeting information and also can see where anyone else is pointing. This paper examined our experience with our first Colab meeting tools. We argued that these tools need to relax the strict application of WYSIWIS to accommodate important interactions in meetings, especially those involving the different and competing needs of individuals, subgroups, and the full group. Relaxations to WYSIWIS were categorized as constraints on space, time, population, and congruence.

We introduced several basic mechanisms for relaxing WYSIWIS in the design of Boardnoter, a meeting tool that most resembles a chalkboard and which has no process model for meetings. We then considered Cognoter, a tool for organizing ideas for a presentation. Cognoter supports a richer and more formal meeting process, which we have used in the Colab for several months. Several problems with Cognoter's ad hoc approach to WYSIWIS were identified.

Two approaches for organizing display space and systematizing the application of WYSIWIS were presented: one based on WYSIWIS stampsheets and the other on multiuser extensions of the Rooms approach developed by Henderson and Card [7]. Our separation of ideas for multiuser interfaces into two competing approaches was for the pedagogical purposes of helping us uncover the underlying tensions and principles. By no means are WYSIWIS Rooms and stampsheets mutually exclusive: actual systems would benefit by combining them.

In considering these proposals, we introduced several concepts for meeting tools and multiuser interfaces besides WYSIWIS, WYSIWIS stampsheets, and WYSIWIS Rooms. One important notion was the idea of lightweight collaborations—collaborations that are formed quickly in the context of a meeting and which persist for just a few minutes. Supporting lightweight collaborations was an important goal for our meeting tools. They are an important source of potential parallelism in computer-supported meetings.

We argued that meeting tools need to model subgroups explicitly and to provide services that respect subgroup boundaries. In the stampsheet approach, we suggested that subgroup boundaries can be determined by an analysis of window use; in the room model, subgroup formation follows explicit user action in which subgroups are assigned their own conference rooms. In both cases we argued the need for a variety of mechanisms to support the changing needs of information access as process transitions occur, as subgroups are formed and dissolved, as individuals shift their focus of activity, and as the group shifts from multiple parallel activities to a single focused activity and back again.

Early observations of Cognoter use have shown that, for lightweight collaborations, subgroups are transient, and it is important to be able to get an overview

of what other subgroups are doing. To this end, we proposed miniaturization as a way of compressing images, both for Rooms and in stampsheets. In both cases we extended the concept of static icons in ways that allow participants to get an overview of the activities. The extensions include active icons, as well as flashing activity indicators, subgroup telepointing indicators, and cumulative change indicators. These mechanisms relax the constraints of strict WYSIWIS to help participants adjust the degree to which they participate in various parts of the discussion and allow a kind of progressive disclosure for scanning parts of the discussion for changes.

Telepointing is an example of one of the new concepts that arise when interfaces are extended to support multiple users in meeting situations. Another example is teleselection, the indication to collaborators of those objects on which an editing operation is being performed. Teleselection recognizes that certain joints in the operations on public information must be made visible in order to admit comment. Mechanisms for displaying teleselection must recognize that multiple activities (including private ones) can be occurring simultaneously, and therefore WYSIWIS must be relaxed to disambiguate the display.

The WYSIWIS stampsheet and WYSIWIS Rooms approaches present us with several trade-offs. Our preliminary hypothesis is that the next generation of Cognoter design will need to pick a combination of features from these approaches that provides low overhead for lightweight collaborations. We also predict that the Rooms approach will scale better for meetings and processes associated with larger groups.

There has been much less experience with multiuser interfaces than with single-user interfaces. This paper represents experience with our first generation of meeting tools and presents speculations about design issues for the next generation. In the next phase of our work, there will be an opportunity to test some of these hypotheses and revisions experimentally.

ACKNOWLEDGMENTS

The authors wish to thank John Seely Brown for creating in PARC's Intelligent Systems Laboratory a rich, multidisciplinary environment in which projects like the Colab can flourish, and Austin Henderson and Stuart Card for reading and critiquing early drafts of this paper.

REFERENCES

1. BLY, S. A., AND ROSENBERG, J. K. A comparison of tiled and overlapping windows. In *Human Factors in Computing Systems: CHI'86 Conference Proceedings* (Boston, Apr. 13-17, 1986), pp. 101-106.
2. CARD, S. 'Windows': Why they were invented, how they help. *The Office* (Mar. 1985), 52-54.
3. CARD, S., PAVEL, M., AND FARRELL, J. E. Window-based computer dialogues. In *Proceedings of the IFIP Conference on Human-Computer Interaction—INTERACT-'84*. North-Holland, Amsterdam, 1985, pp. 239-243.
4. DONAHUE, J., AND WIDOM, J. Whiteboards: A graphical database tool. *ACM Trans. Off. Inf. Syst.* 4, 1 (Jan. 1986), 24-41.
5. FOSTER, G., AND STEFIK, M. Cognoter, theory and practice of a colab-orative tool. In *Proceedings of the Conference on Computer-Supported Cooperative Work* (Austin, Tex., Dec. 3-5), 1986.

6. GREIF, I., SELIGER, R., AND WEIHL, W. Atomic data abstractions in a distributed collaborative editing system (extended abstract). In *Proceedings of the 13th Annual ACM Symposium on Principles of Programming Languages* (St. Petersburg Beach, Fla., Jan.). ACM, New York, 1986, pp. 160-172.
7. HENDERSON, D. A., AND CARD, S. K. Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. Intelligent Systems Laboratory, Tech. Rep., Xerox Palo Alto Research Center, Palo Alto, Calif., July 1986.
8. PANKO, R. R. Office work. *Office: Technol. People* 2, 1964, 205-238.
9. SCHNEIDERMAN, B. Direct manipulation: a step beyond programming languages. *IEEE Comput.* (Aug. 1983), 57-69.
10. STEFIK, M., FOSTER, G., BOBROW, D. G., KAHN, K., LANNING, S., AND SUCHMAN, L. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Commun. ACM* 30, 1 (Jan. 1987), 32-47.

Received November 1986; revised December 1986; accepted February 1987