

Groupware Experiences in Three-Dimensional Computer-Aided Design

Li Shu

MIT, Room 3-443
77 Massachusetts Avenue
Cambridge, MA 02139
(617) 253-2349 lishu@athena.mit.edu

Computer-Aided Design Laboratory
Department of Mechanical Engineering
Massachusetts Institute of Technology

Woodie Flowers

MIT, Room 3-461a
77 Massachusetts Avenue
Cambridge, MA 02139
(617) 253-6234 flowers@athena.mit.edu

ABSTRACT

A system that allows people to simultaneously modify a common design in a graphically rich environment was developed to identify and examine groupware interface issues unique to three-dimensional computer-aided design.

Experiments confirmed that a simultaneous mode of edit access is preferred over a turn-taking mode for two-person interactions. Also, independent points of view (e.g., isometric versus top view) between designers optimized parallel activity. Further experiments that aimed to transfer software-usage knowledge through the groupware system led to the development of the *viewpoint*. The viewpoint is a tool that indicates the points of view of different designers as well as provides a method of pointing effective in an environment where arbitrary, contrasting points of views are allowed.

KEYWORDS

Computer-aided design, graphics, three-dimensional.

INTRODUCTION

Motivation

Many people realize the impact of the human-computer interface on the success of a software product [14][3][4]. The significance of the human-computer interface extends naturally to groupware products where interface-design choices are even more complex. Moreover, [9] asserts that many of the perceptions and perspectives acquired from experience with single-user applications will not be helpful for the groupware domain. Unfortunately, only recently have there been interface research on group systems, which

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.

© 1992 ACM 0-89791-543-7/92/0010/0179...\$1.50

involve not only the human-computer interaction existing in a single-user application, but also the person-to-person interaction through the groupware system [13]. Much of this work has focused on primarily text editors that support group creation and modification of documents and software code [10][12][8][15]. One of the few systems that specifically supports computer-aided design (CAD) for example, is TOPES [18]. Studies that involve the collaborative use of a two-dimensional drawing surface include [2] and [24].

Not surprisingly, the existence of "profound occupational and suboccupational differences in the way in which workgroups share space and structure activities in their work environments" has been documented:

"...we observe that some occupational groups, such as artists, architects and mechanical engineers (designers of physical objects whose development is shared in posted drawings or sketches), tend to prefer open workspaces through which colleagues are encouraged to browse. Other occupational groups (e.g., software engineers, academics, writers) tend to prefer more enclosed and private workspaces which offer fewer intrusions and interruptions." [19].

This suggests that designers and authors need systems with inherently different orientations to best support their work; an editor which tolerates graphics capabilities may not be sufficient for supporting the graphics activities of collaborative design work. It is then possible that there exist issues unique to a drawing or CAD system that will not be recognized and appreciated in the context of a group editor which primarily supports the manipulation of text. Also, representation of three-dimensional objects may present situations not applicable to systems that only provide a planar sketching surface.

System

Clearly, to study interface issues, an appropriate system is needed. A brief description of Teledesign, the system implemented, follows.

Teledesign was built from scratch, specifically designed to support multiple users, although it can also be used by a single designer as needed. An alternative option is to write a shell program to transform existing single-user applications into multi-user applications, examples of which are SharedX from Hewlett-Packard Co., [11], and [17]. Although this alternative method can take advantage of existing software, designing a groupware application from the ground up provides an interface more suited for group use. Teledesign supports *real-time*, physically dispersed as well as face-to-face meetings of several people using a *replicated architecture*. In a replicated architecture, a local copy of the application, the graphics editor in this case, runs at each node of the conference. Inputs from designers at each node are broadcasted to all other nodes where the outputs are determined and executed locally. This option is in contrast to a centralized architecture, where only one application accepts input from the nodes and then determines and broadcastes all resulting output. Benefits of the replicated architecture over the centralized architecture are identified in [6]. Teledesign supports both *sequential access*, where only one designer can input edits at a time, and *simultaneous access*, where all the designers can edit at the same time. In sequential edit-access mode, an *implicit floor change policy* is implemented, where designers do not have to perform explicit actions such as selecting a particular button or key to request and relinquish floor control. During simultaneous access, *reversible execution* [20] was used for concurrency control. Reversible execution involves executing commands locally, before they are broadcasted, in a way such that they can be undone if they lead to inconsistencies between databases. The taxonomy used above is delineated in [21] and [7].

Teledesign has two main parts. A graphics editor runs at each workstation connected to the conference, and a communication system propagates changes between the workstations. The graphics editors are connected to a *hub*, which uses UNIX interprocess communications sockets to receive and distribute updates. The hub follows the client-server model of interaction, and accommodates connections across different remote networks as easily as those on the same network. The three-dimensional graphics editor, implemented on Silicon Graphics Personal Irises, provides a user-friendly and graphically rich environment where designed objects are *built* out of blocks and cylinders of various colors and shapes.

The Teledesign graphics editor is simple. Difficulties with explaining a previous, more complex editor to potential users resulted in streamlining which led to the current editor. In the current Teledesign editor, there is only one window. This avoids a sense of clutter [23] and reduces the computational load of redrawing many complex surfaces in multiple windows. Although some more experienced users

have expressed a preference for simultaneous display of multiple views, a single window was chosen both to avoid overwhelming the new user and to provide a larger workspace. In this window, a single view of the color-shaded objects is displayed. A different view is obtained by using special keyboard keys to navigate about in the virtual world of the objects. A set of axes whose origin coincides with the origin of the world may be toggled on and off. It was found that it is easier to manipulate an object in a three-dimensional world using a two-dimensional input device, such as the mouse, by being able to *snap* to and work from views perpendicular to one of the three reference frame axes (see Figure 1). A single pop-up menu is used to create, duplicate, or destroy a cylinder or rectangular block. Setting the material or color for subsequent creations is also accomplished using the menu. Editing functions to translate, rotate, and change the shape of the cylinders or blocks, were made as intuitive as possible by using mouse click-and-drag operations. A detailed description of the system and available functions can be found in [22].

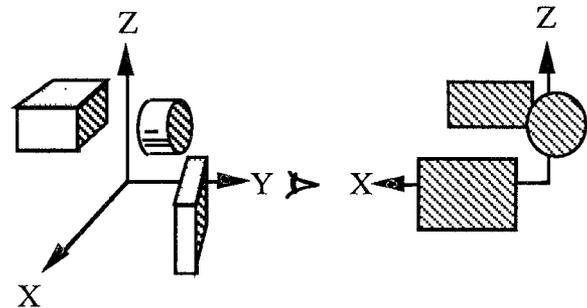


Figure 1. Perspective View vs. Orthogonal "Snap" View.

Experiments in which two-person groups performed design tasks cooperatively were conducted on Teledesign. The first set of experiments examined the issues of floor-control passing and display-consistency enforcement, while the second paired designers of contrasting Teledesign-usage experience in order to transfer software-usage knowledge through the groupware system.

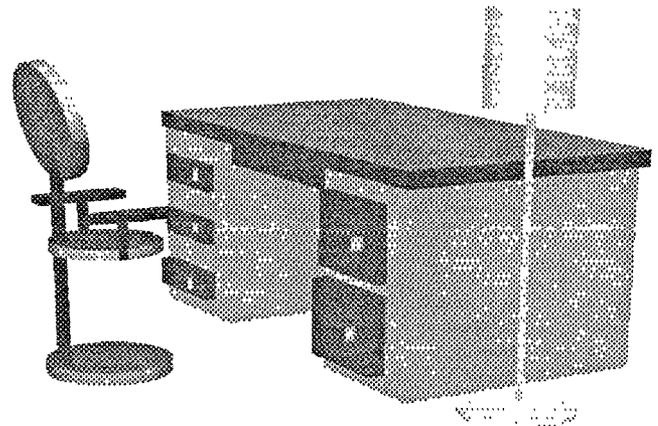


Figure 2. Product of an Experimental Collaboration Session on Teledesign.

EXPERIMENT 1

A useful metaphor for Teledesign is a virtual meeting place for designers collaborating from afar. In this virtual meeting place, the designers are gathered around a set of objects, representing a prototype of a design, and with minimal physical effort, the meeting participants would modify or build upon this prototype.

Questions

Should everyone be able to manipulate the objects simultaneously or should they have to take turns? What would be the appropriate protocol for taking turns? This is a question common of all groupware systems: what, if any, edit-access or floor-passing protocol should be implemented?

Another common interface-design issue in groupware involves the degree of sharing. [23] introduces WYSIWIS (What You See Is What I See), the foundational abstraction for multi-user applications that embodies many of the attributes of a face-to-face meeting. A strict enforcement of WYSIWIS produces an environment where everyone sees the same thing all the time. One dimension along which WYSIWIS can be relaxed is congruence, which results in allowing alternative views, or "visual variations" such that a remote selection can be displayed differently from a local selection. [7] defines a *shared context* as "a set of objects where the objects and the actions performed on the objects are visible to a set of users," and a *view* as a "visual representation of some portion of a shared context." The same information may be presented differently in different views, or different sections of the shared context may be viewed using the same presentation format. For example, an array of numbers can be shown as either a table or a graph, and different people may be looking at different sections of the table [7].

To further illustrate the difference between a shared context and a shared view, particularly in the Teledesign environment, return to the meeting metaphor where designers are manipulating a shared design prototype. When one person modifies the set of objects, her actions are seen by everyone else. This group of people has a shared context since an identical set of objects is discussed at all times. Yet, since each person has a different point of view of the objects depending on where the person is standing or seated, this group does not have a shared view of the context. What degree of sharing, or WYSIWIS, is most appropriate? Should all the computer screens look identical all the time?

Methods

In this experiment, the effects of two edit-access modes, or "meeting styles", a free-for-all, simultaneous mode and a simple turn-taking mode were observed on the performances of two types of collaborative design tasks. The first type of task was a loosely defined, loosely linked task, an example of which is the cooperative design of a room. Much freedom is given to the designers, and what one person does at one end of the room usually does not substantially affect

the other. The other type of task was a well-specified interdependent task, such as the collaborative building of a bookcase. This was a more interdependent task because the dimensions of one panel may depend on the dimensions of other panels, which may be designed by different people.

Research Setting

Two SGI Personal Irises located side-by-side were used for the experiments. A partition was placed between them so that the participants could see neither the other person nor the screen of the other person. Although they were visually separated, the participants were allowed to talk freely. All the sessions were videotaped.

Subjects and Group Task

This experiment used five groups of two each. The participants were MIT and Tufts graduate students in engineering who were unfamiliar with the program. They were trained in a structured half-hour session immediately prior to the experiment. In the training session, the participants first familiarized themselves with commands of the editor in a single-user environment by performing simple tasks, such as the creation and subsequent manipulation of blocks and cylinders into specified shapes and orientations. The single-user environments were achieved by executing two hubs, and connecting each graphics editor to a different hub. Next, they performed a similar task cooperatively to become accustomed to the presence of another designer in the same workspace.

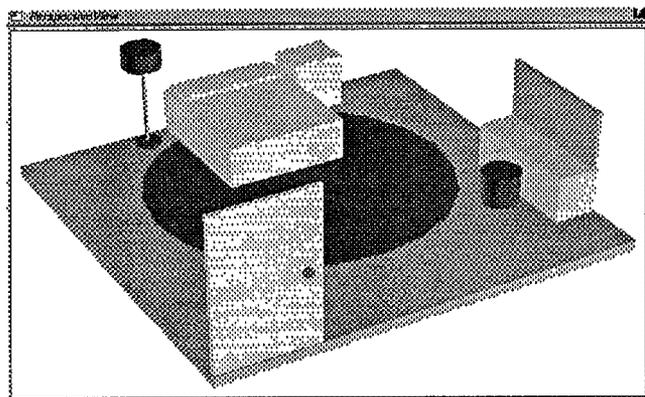
The goals of the experimental tasks were to collaboratively design objects and sets of objects using the cylinder and rectangular block as primitives. Each of the two edit-access modes was paired with each of the two types of tasks as described above, so that each group performed a total of four tasks, a well-specified, interdependent task under both the simultaneous and sequential modes, and a loosely linked, loosely specified task under both the simultaneous and the sequential modes. Two different tasks of the same type were performed under each of the floor-passing modes since the subjects would have gained familiarity with an identical task performed twice. All four tasks were completed in approximately one-and-a-half to two hours.

Observations

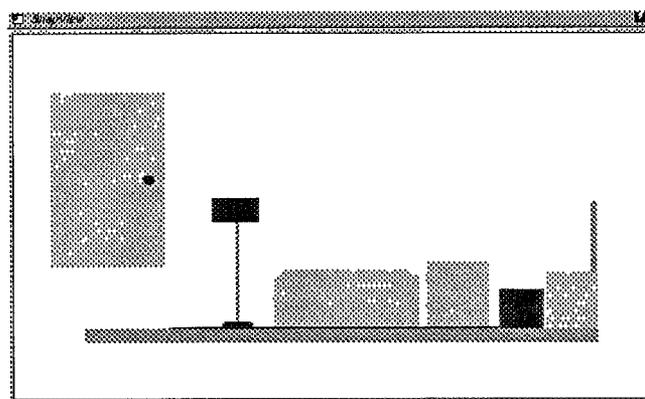
Independent Environments

It was found that independent environments were necessary to optimally perform independent parallel tasks, especially during the simultaneous edit-access mode. One example involves the status of *edit-mode*. When edit-mode is toggled on, the shape of objects can be changed by selecting the corresponding highlighted outlines, or *edit-lines*. During edit-mode, objects can also be translated using the same mouse button when edit-lines are not selected. Unfortunately, it is difficult to avoid the edit-lines of a very small object. Consequently, users turn edit-mode off to translate a small object without inadvertently changing the size. This results in a conflict when the edit-mode status is common across workstations, and one designer is editing shapes while the other wants to move small objects.

Another example of the need for independent environments involves points of view. When the designers are working on separate tasks, it is intuitive that they prefer to be in the view that best accommodates their task. For example, someone rearranging the floor plan of the room would work best from the top view while the designer modifying the height of a door would work best from a side view. Additionally, it was observed that having different points of view is beneficial even when designers are working cooperatively on a single task. For example, when one designer is maneuvering an object, the other can see the position of the object, which as shown from another view, is not apparent to the first designer from his view (see Figure 3).



a. Non-aligned Perspective View of First Designer.



b. Same Geometry, Aligned Snap View of Second Designer.

Figure 3. Misleading Apparent Object Position Due to Point of View. (Note position of door in both views.)

[20] recognized the need for presenting the same data in different ways when participants have varied interests. [23] found that the broadcast of a participant's input command to modify an environment parameter, such as the changing of the position of a public window, would result in obstructing private work on other screens. Such conflicts were resolved by relaxing the congruence constraint of WYSIWIS to allow independent environments.

Yet when people were allowed to have independent views, a need for common reference points ensued. One of the most

commonly asked question between designers was, "Where are you?" or "What view are you in?" This suggested a need for a representation of other participants' views in each designer's screen.

Edit-Access Style

It was found that forced turn-taking was not necessary for two-person interactions, regardless of task type. Contrary to intuition, the confusion caused by simultaneous editing was minimal. Occasionally, after some pause in verbal communication, two people would try to grab and move the same object without being aware of the other person's actions. When this happens, the designers need to first recognize that they are trying to move the same object, and then they need to resolve the conflict. During this experiment, the total time spent both on recognizing and resolving such conflicts was on the order of five seconds per fifteen minutes. The appropriate meeting metaphor is that there is no need for a moderator for two-person meetings. When turn-taking is necessary, it will usually occur naturally out of social protocol. [7] also found that simultaneous editing is not chaotic due to intervention of social protocol, and that, "Collisions are surprisingly infrequent." To further prevent accidental collisions, a "busy signal" such as the graying of teleselected objects, or the use of a cloudburst around changing text, has been implemented in text editors to warn other participants not to modify the same data [23] [7]. Likewise, if the virtual positions of other designers in the meeting room were somehow represented in each workstation screen, people would more likely avoid other designers' workspaces.

Furthermore, the experiment participants unanimously preferred the simultaneous edit-access mode over the sequential edit-access mode for both types of tasks. The designers found it frustrating to be forced to take turns when parallel work was possible. [7] also recognized that the sequential-access mode is "limited to those situations in which a single active user fits the dynamics of the session, (and that) it is particularly ill-suited for sessions with high parallelism, inhibiting the free and natural flow of information." [25] reports that the type of floor mode selected depends on the type of meeting, noting that when persons of equal rank met, the "baton mode and first-come-first-served mode" were used most frequently, the "designation mode" was used when a clear chairperson was present, and that brainstorming sessions used the "free mode" most often. [1] presents additional factors that affect patterns of control in a computerized meeting room. [20] also favors less structured methods of access control, finding that "participants in a real-time conference change roles quite frequently, and complex access controls may be a hindrance." [20] promotes verbal negotiations as an alternative to a rigid floor-passing scheme, admitting however, that verbal negotiation may break down when the size of the meeting group increases since "the voice channel itself becomes an object of contention." Returning to the meeting metaphor, when the size of the meeting increases, more structure may be needed for the meeting to remain effective.

EXPERIMENT 2

The use of groupware to gain software knowledge has been cited by many. [4] points to the need to support informal collaboration to learn how to use the technology, further suggesting that the system help match learners with experts. [18] categorizes three areas of interaction of TOPES, which is an interactive graphics teleconferencing system used for the planning of building construction. One area was related to the discussion of drawings and the exchange of technical information, while the other two were related to the training of new operators and the deployment of the systems. Novice TOPES users who came upon difficulties would set up conferences with more experienced persons. [18] affirms, "The teleconferencing feature has facilitated user mastering of the command language."

Methods

In this experiment, designers of contrasting levels of Teledesign-usage experience were paired. The goal of the task was to transfer knowledge of software use across the system to a person who has never been exposed to Teledesign. This situation is different from the first set of experiments, which paired designers of comparable expertise, and may reveal implications for transferring more general technical, or other, knowledge across the groupware system. Again the participants were visually separated but allowed to talk freely. First the system expert steps the novice through each of the available functions, demonstrating how to best take advantage of the system features. Next, the team performs a simple collaborative design task suggested by the novice, which lasts approximately half an hour. Examples of designs selected include those of a desk and chair set, a wheelchair, a trailer, and various room layouts.

Observations

Pointing

Pointing was important. The need for a *telepointer*, a pointer that can be seen on multiple displays, was more pronounced in the context of this experiment than in the first. The need to reference objects and parts of objects specifically occurred more frequently than during experiment one. An example of a typical verbal exchange is, "To change the shape of *this* block in *this* dimension, it would easiest to select *this* corner, while editing from *this* view." Designers who were new to the system assumed that their cursor was visible in the other's display. Unfortunately, pointing in three dimensions is more complex than sending the two-dimensional cursor location on the computer screen from one workstation to another. Two cursors in identical positions relative to the workstation screen may be pointing to completely different objects if the two workstations are displaying the objects from different points of views. [23] came across the two-dimensional analogy of this telepointer problem when individualized preferences for window placements were allowed, which was solved by a calibration to make the telepointer window-relative rather than screen-relative. There is no equivalent solution for pointing in three dimensions. The participants of the experiment improvised by actually "picking up" and

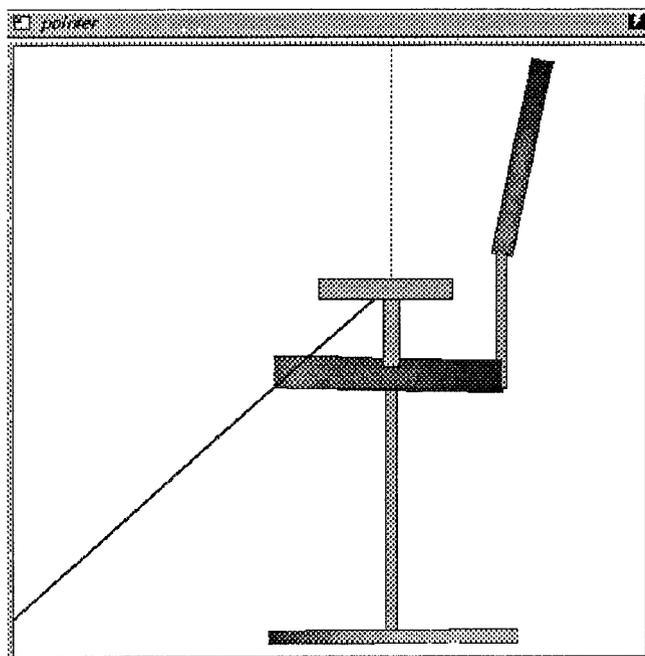
jiggling an object. This method has the disadvantage of disturbing the position of the referenced object. A less intrusive method involved creating a makeshift pointer out of an appropriately dimensioned block, and then moving the block to the vicinity of the object being discussed. Still, this method is guaranteed to work only if both designers have identical points of view.

Point of View

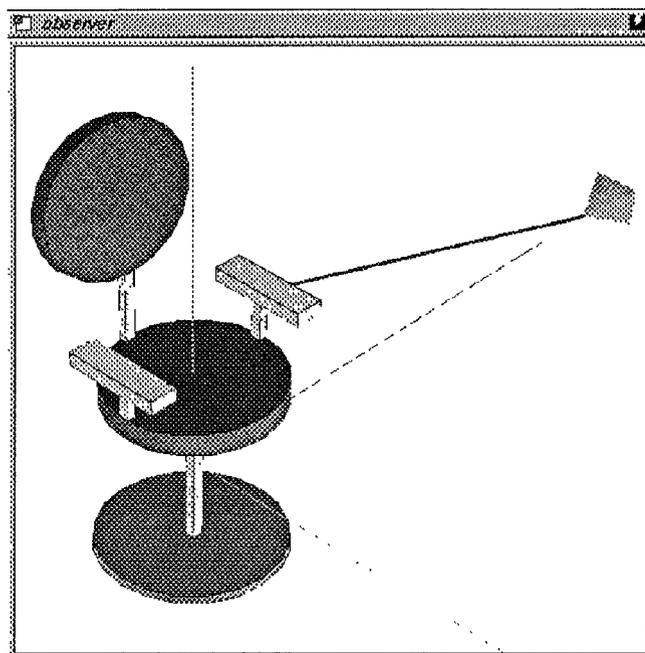
Knowledge of the other person's point of view was more essential in this situation than it was in experiment one. Since designers were allowed independent points of view, it was difficult to convey the concept that certain things were easier to do from certain points of view. The novice, not being able to see the more experienced user change between views to accomplish particular tasks, tended to try to do everything from an unaligned perspective view, only to become frustrated in the process. It was also difficult for the system expert to recognize that the other person was having difficulties because he was editing in a view which was not best suited for the task.

[16] suggests that in training simulations, instructors and students may want to have identical views to share both context and view. [11] suggests the option of "snapping back" to a common view. [23] "teleports" people to view the appropriate windows. It would seem that enforcing strict WYSIWIS is a simple way of solving both problems: there would be no confusion about point of view, and pointing reduces to the two-dimensional case. In fact, strict WYSIWIS, such that the position of cursors relative to objects and menus are visible across workstations with identical views, may be the most efficient mode for several training tasks, especially when there is a language barrier or other communication hindrance. However, sudden changes in point of view can be disorienting, especially when the change is not self-initiated [7], just as "scrollbar wars" are disruptive [23]. Perhaps voluntary teleporting, where users can choose to align their views, as suggested in [20] would be more acceptable. Alternatively, [7] identifies real-time animation as a technique of smoothly reflecting changes. The example given involves having text materialize gradually or change color as it is entered. Animation use in Teledesign could involve the more continuous panning of views to get from one point of view to another, such that the transition is less abrupt. Nevertheless, animation is computationally expensive and may jeopardize system responsiveness.

Perhaps the idea of displaying representative vectors of all other designers' views in each person's window, as suggested in results of experiment one, would be a more effective way of conveying how and when which views are used by other participants. Still, if designers continue to have independent points of views, an effective method of pointing which would accommodate the different views would have to be developed.



a. View of Designer Who is Pointing.



b. View of Designer Who is Observing.

Figure 4. Implementation of Viewpoint.

Solution

In answer to both the point-of-view and the three-dimensional pointer issues, the *viewpoint* was implemented. A viewpoint is a pyramid that represents the view of a designer and is visible in all other designers' displays. When a designer wants to point, the pointer is turned on using the pop-up menu, after which a gray line, extending from a lower corner to the center of the screen, appears to show that the pointer is available and ready. When an object is selected, the line brightens and

terminates on the corresponding object. In the views of observing designers, a ray runs between the tip of the viewpoint representing the designer who is pointing and the selected object (see Figure 4). Local pointers are colored differently from remote pointers.

A possible feature of the viewpoint that had not been implemented, is to be voluntarily teleported to the view of another designer by selecting the corresponding viewpoint. A way of returning quickly to the original position, much like the "backdoor" of [23] should be provided.

One advantage of the viewpoint is that it serves its purpose without being unnecessarily distracting, at least no more distracting than the presence or movement of another designer in a real-world room. [7] supports that "an active user is only marginally interested in others' changes, which should therefore be indicated subtly and not disruptively." Implementation of the viewpoint is not only less disruptive, but also computationally cheaper than opening additional windows to show particular views, since only one additional object is displayed per designer in an existing window. Proper scaling of the viewpoint makes it easy to recognize if someone else has the same point of view without being an obstruction to the view itself. When the pointer is used, it does point to the correct object effectively regardless of the view of any particular designer. Alternatively, a telepointer in the form of a three-dimensional arrow has a higher chance of being obstructed in another designer's display that is in a different point of view.

SUMMARY

Teledesign was built to investigate human-machine interface situations particular to cooperative work in three-dimensional computer-aided design. Experiments confirmed that a two-person collaborative effort called for a simultaneous rather than a forced turn-taking mode of edit access. It was also found that allowing designers to have independent points of view optimized parallel activity when independent tasks were performed, as well as assisted the designer with crucial feedback from another participant in a different perspective. The need for both a three-dimensional pointer and the knowledge of the other designer's point of view was established. Consequently, the viewpoint was developed as a tool that indicates the points of view of different designers and provides a method of pointing which is effective in an environment where arbitrary, contrasting points of views are allowed.

Clearly, much territory remains to be explored. For one, Teledesign has not been tested for groups of more than two persons. An increase in group size may uncover entire realms of new issues. As the group size expands, the degree of active participation of the average meeting member may reduce, thereby intensifying the need to access private work.

A larger group size mandates a system capable of accommodating the various needs of many people. Time is

one dimension along which the system can be made more flexible. Since it is difficult to schedule a meeting time for several people, participants may wish to drop in and out during the course of a meeting. [16] recognized the need to update a late-joining user. [19] recommends that a system support both synchronous and asynchronous modes of interaction since this represents the "natural patterns of communication" of workgroup members. [18] supports asynchronous interaction in addition to real-time teleconferencing by allowing collaborators to send messages and transfer drawings. [5] describes the Design Journal, a hypertext system that captures and documents design rationale. A "history file" had been implemented as part of an earlier version of Teledesign. The history file records edits of all designers in chronological order, and identifies which edits were performed by whom. When a late comer, or even someone who has entirely missed the meeting, logs on, he is presented with the current state of objects, and has the option of running through the history file to witness the edits which resulted in the current state. A combination of the history file and the Design Journal may be implemented as a voice-annotated, animated graphical segment, much like a videotape segment, which explains the rationale behind the changes as they are replayed.

ACKNOWLEDGMENTS

Financial support for this research was provided in part by the Office of Naval Research through an academic fellowship.

REFERENCES

1. Austin, L.C., Liker, J.K., and McLeod, P.L. Determinants and Patterns of Control over Technology in a Computerized Meeting Room, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 39-52.
2. Bly, S.A. A Use of Drawing Surfaces in Different Collaborative Settings, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Portland, OR, September 26-29, 1988), ACM, New York, pp. 250-256.
3. Bullen, C.V. and Bennett, J.L. Learning from User Experience with Groupware, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 291-302.
4. Clement, A. Cooperative Support for Computer Work: A Social Perspective on the Empowering of End Users, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 223-236.
5. Conklin, J. and Begeman, M.L. gIBIS: A Hypertext Tool for Exploratory Policy Discussion, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Portland, OR, September 26-29, 1988), ACM, New York, pp. 140-152.
6. Crowley, T., Milazzo, P., Baker, E., Forsdick, H., and Tomlinson, R. MMConf: An Infrastructure for Building Shared Multimedia Applications, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 329-342.
7. Ellis, C.A., Gibbs, S.J., and Rein, G.L. Groupware, Some Issues and Experiences. *Communications of the ACM*. Vol. 34, No.1, (January 1991), pp. 38-58.
8. Galegher, J. and Kraut, R.E. Computer-Mediated Communication for Intellectual Teamwork: A Field Experiment in Group Writing, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 65-78.
9. Grudin, J. Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Portland, OR, September 26-29, 1988), ACM, New York, pp. 85-93.
10. Kaiser, G.E., Kaplan, S.M., and Micallef, J. Multiuser, Distributed Language-Based Environments, *IEEE Software*, Vol. 4., No. 6 (November 1987), pp. 58-67.
11. Lantz, K.A. An Experiment in Integrated Multimedia Conferencing, *Proceedings of the Conference on Computer-Supported Cooperative Work* (Austin, TX, Dec. 3-5, 1986), ACM, New York, pp. 267-275.
12. Leland, M.D.P., Fish, R.S., and Kraut, R.E. Collaborative Document Production Using Quilt, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Portland, OR, September 26-29, 1988), ACM, New York, pp. 206-215.
13. Mantei, M. Capturing the Capture Concepts: A Case Study in the Design of Computer-Supported Meeting Environments, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Portland, OR, September 26-29, 1988), ACM, New York, pp. 257-270.

14. Martin, J. *Design of Man-Computer Dialogues*, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.
15. Neuwirth, C.M., Kaufer, D.S., Chandhok, R., and Morris, J.H. Issues in the Design of Computer Support for Co-authoring and Commenting, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 183-196.
16. Patterson, J.F., Hill, R.D., and Rohall, S.L. Rendezvous: An Architecture for Synchronous Multi-User Applications, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 317-328.
17. Perreault, D.J. *Interactive Computer Support for Remote Design Teams: A New Approach*, MIT S.M. thesis, Cambridge, MA, 1991.
18. Pferd, W., Peralta, L.A. and Prendergast, F.X., Interactive Graphics Teleconferencing, *IEEE Computer*, Vol. 12, No. 11, (November 1979), pp 62-72.
19. Reder, S. and Schwab, R.G. The Temporal Structure of Cooperative Activity, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 303-316.
20. Sarin, S. and Greif, I. Computer-Based Real-Time Conferencing Systems, *IEEE Computer*, Vol. 18, No. 10, (October 1985), pp. 33-45.
21. Schooler, E.M., Casner, S.L., and Postel, J. Multimedia Conferencing: Has It Come of Age? *Proceedings of the 24th Hawaii International Conference on System Sciences*, Vol. 3, (January 1991), pp. 707-716.
22. Shu, L. *Groupware Experiences in Three-Dimensional Computer-Aided Design*, MIT S.M. thesis, Cambridge, MA, 1992
23. Stefik, M., Bobrow, D.G., Foster, G., Lanning, S., and Tatar, D. WYSIWIS Revised: Early Experiences with Multiuser Interfaces, *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, (April 1987), pp 147-167.
24. Tang, J.C. and Leifer, L.J. A Framework for Understanding the Workspace Activity of Design Teams, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Portland, OR, September 26-29, 1988), ACM, New York, pp. 244-249.
25. Watabe, K., Sakata, S., Maeno, K., Fukuoka, H., and Ohmori, T. Distributed Multiparty Desktop Conferencing System: MERMAID, in *Proceedings of the Conference on Computer-Supported Cooperative Work* (Los Angeles, CA, October 7-10, 1990), ACM, New York, pp. 27-38.