Collaborative Computing, **1** (1994) 1–14

Teledesign: Groupware user experiments in three-dimensional computer-aided design

LI SHU and WOODIE FLOWERS

Computer-Aided Design Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Received January 1993

Teledesign is a groupware system that allows people to modify a common design simultaneously in a graphically rich environment. This system was developed to identify and examine groupware interface issues unique to three-dimensional computer-aided design. Experiments have revealed that a simultaneous mode of edit access is preferred over a turn-taking mode for the two-person collaborative design tasks observed. Independent points of view (e.g. isometric compared with top view) between designers also optimizes parallel activity. Further experiments that aimed to transfer software-usage knowledge through the groupware system led to the development of the view-point. The viewpoint is a tool that indicates the points of view of different designers and enables three-dimensional pointing for an environment where designers have arbitrary, contrasting points of view.

Keywords: Groupware, CSCW, computer-aided design, three-dimensional workspace, user interface

1. Introduction

1.1. *Motivation of concurrent engineering*

The traditional, serial manner of product development has been found to result in difficulties such as long development times and designs that are difficult to manufacture [1]. Serial development has been described as a sequence of 'throw it over the wall' processes, where one functional group, product design for instance, makes its contribution to a project independently of other groups. The project is then 'tossed over the wall' to the 'downstream' group, production operations possibly, who must work with decisions that were made without its input but considerably affect the groups objectives.

Clausing [1] asserted that the consequent waste of time and poor communication between functional groups would be decreased if a multifunctional product development team were implemented. This team would be comprised of representatives from various groups such as product design, production-process engineering, field-support development, and marketing, and would aim to execute the concurrent product development process in a seamless manner. As members of the product development team are likely to be situated in geographically disparate locations, appropriate groupware has the potential to support the activities of this team.

1.2. *Preview*

Teledesign, a system that enabled multi-user simultaneous editing of graphical representations, is described. *Teledesign* was developed as a platform to explore the use of groupware for collaborative design tasks. Two sets of user experiments were performed to uncover issues relevant to group editing of three-dimensional graphical objects.

1.3. *Importance of human-computer interface*

Many people have realized the impact of the human-computer interface on the success of a software product [2–4]. The significance of the human-computer interface extended naturally to groupware products where interface-design c oices are even more complex. Moreover, Grudin [5] asserted that many of the perceptions and perspectives acquired from experience with single-user applications would not be helpful for the groupware domain, as interface research on group systems would involve not only the human-computer interaction existing in a single-user application but also the person-to-person interaction through the groupware system [6]. Kiesler *et al.* [7] reported social psychological aspects of computer-mediated communication between people. Much of the interface research on group systems has primarily focused on text editors that support group creation and modification of documents and software code [8–11]. One of the few systems that specifically supported computer-aided design (CAD) is TOPES [12]. Studies that involved the collaborative use of a two-dimensional drawing surface included Bly [13] and Tang *et al.* [14].

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." [15]

This suggested that designers and authors needed systems with inherently different orientations to best support their work; an editor that tolerated graphics capabilities might not be sufficient for supporting the graphics activities of collaborative design work. It was then possible that there were issues unique to a drawing or CAD system that would not have been recognized and appreciated in the context of a group editor which primarily supported the manipulation of text. It was also possible that representation of objects in three dimensions would present situations not applicable to systems that provided a planar sketching surface.

1.4. *Teledesign system*

A brief description of *Teledesign*, the system implemented to study interface issues, follows. *Teledesign* was built from scratch, specifically designed to support multiple users, although it could also have been used by a single designer as required. An alternative option was to write a shell program to transform existing single-user applications into multi-user applications, examples of which were SharedX from the Hewlett-Packard Co. [16, 17]. *Teledesign* supported 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, ran at each node of the conference. Inputs from designers at each node were broadcast to all other nodes where the outputs were determined and executed locally. This option was in contrast to a centralized architecture, where only one application accepted input from the nodes and then determined and broadcast all resulting output. Benefits of the replicated architecture over the centralized architecture were identified by Crowley *et al.* [18]. *Teledesign* supported both sequential access, where only one designer could enter edits at a time, and simultaneous access, where all the designers could edit at the same time. During the sequential edit-access mode, an implicit floor change policy was implemented, where designers did 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 [19] was used for concurrency control. Reversible execution involved executing commands locally, before they were broadcast, in a way such that they could be undone if they led to inconsistencies between databases. The taxonomy used above was delineated by Schooler *et al.* [20] and Ellis *et al.* [21].

Teledesign had two parts. A graphics editor ran at each workstation connected to the conference, and a communication system propagated changes between the workstations. The graphics editors were connected to a hub, which used UNIX interprocess communications sockets to receive and distribute updates. The hub followed the client-server model of interaction, and accommodated connections across different remote networks as easily as those on the same network. The three-dimensional graphics editor, implemented on Silicon Graphics Incorporated Personal Irises, provided a user-friendly and graphically rich environment where designed objects were built out of blocks and cylinders of various colours and shapes.

In the *Teledesign* editor, only one window, which contained a single view of the colour-shaded objects, was displayed per workstation. A set of axes whose origin coincided with the origin of the virtual world of the objects could be toggled on and off. Any view looking towards the origin could be obtained by using special keyboard keys to navigate about in the virtual world of the objects. A single pop-up menu was used to create, duplicate, or destroy a cylinder or rectangular block. Setting the material or colour for subsequent creations was also accomplished using the menu. Editing functions to translate, rotate, and change the shape of the cylinders or blocks, were performed using mouse click-and-drag operations. A detailed description of the system and available functions are available [22].

The *Teledesign* graphics editor was designed for ease of use. Difficulties with explaining a previous, more complex editor to potential users resulted in streamlining that led to the current editor. Although some experienced users had expressed a preference for simultaneous display of multiple views, display of a single view per workstation was chosen both to avoid overwhelming the new user and to provide a larger workspace. The display of a single view per terminal also reduced the computational load of redrawing many complex surfaces in multiple windows, thus increasing the system responsiveness. Casual usage during software development revealed that it was easier to manipulate an object in a three-dimensional world using a two-dimensional input device, such as the mouse, by being able to align with and work from views perpendicular to one of the three reference frame axes (see Fig. 1). Therefore, the editor allowed fast access to orthogonal 'snap' views.

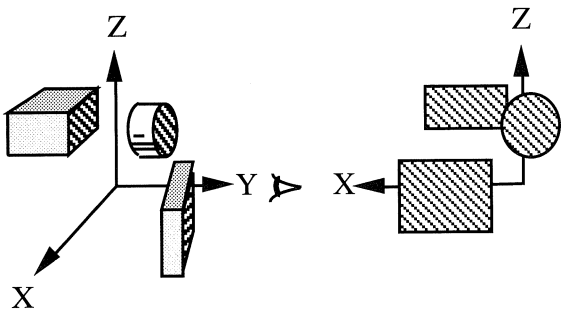


Fig. 1. Perspective view compared with orthogonal 'snap' view.

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

2. Experiment 1: Edit access modes

2.1. *Experiment goals and design*

In the first set of tests, the primary goal was to determine the effects of different modes of edit access, or floor-passing schemes on the performance of different types of collaborative design tasks. Two modes of edit access were implemented on *Teledesign*. One allowed simultaneous access by multiple participants

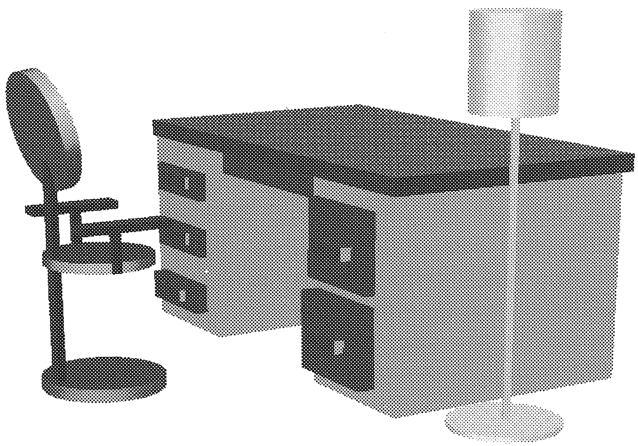


Fig. 2. Product of an experimental collaboration session on teledesign.

to the database. The other forced a sequential mode of access by accepting input from only one editor at a time. These two modes of edit access were observed on the performance of two types of collaborative design tasks. The first type of task was a loosely-defined, loosely-linked task. We chose the cooperative design of the contents and layout of a room to represent this type of task. Much freedom was given to the designers, and what one person did at one end of the room usually did not substantially affect the other. The other type of task was a well-specified, interdependent task, such as the collaborative building of a bookcase, where the dimensions of one panel may depend on the dimensions of other panels, which may be designed by different people. It was expected that enforced turn-taking would be perceived as more of a nuisance for tasks that permitted a higher amount of parallel activity.

2.2. *Research setting*

Two Silicon Graphics Incorporated 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 could talk freely. All of the sessions were videotaped.

2.3. *Subjects and group task*

This experiment used five groups of two subjects. The participants were MIT and Tufts University graduate students in engineering who were unfamiliar with the program. They learned how to use the program in a structured half-hour session immediately prior to the experiment. In the training session, the participants first familiarized themselves with the 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, the subjects performed a similar task co-operatively to become accustomed to the presence of another designer in the same workspace. At this time, no partition was used, so that the participants may gain an idea of the degree of sharing.

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, so that every 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 so that the subjects would not gain familiarity with an identical task performed twice. All four tasks were completed in approximately one-and-a-half to two hours.

2.4. *Observations*

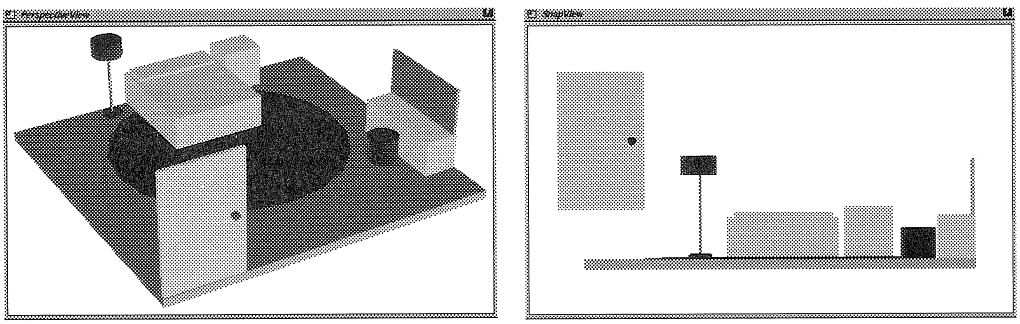
2.4.1. *Independent environments*

During both modes of edit access, only the actions that affected the state of the set of objects were transmitted between workstations. Commands to activate a menu, or change points of view were not broadcast automatically, although it was possible for one designer to send particular points of view to all workstations. This seemed to have been a fortuitous oversight, as it quickly became clear that during normal usage, broadcasting commands to activate menus would be distracting, if not confusing, and enforcing consistent points of view across workstations would only hinder the efficiency of the group task, regardless of the mode of edit access.

The need for independent environments to perform independent parallel tasks optimally, especially during the simultaneous edit-access mode, was reinforced when the status of edit-mode was inadvertently broadcast. When edit-mode was toggled on, the shape of objects could be changed by selecting the corresponding highlighted outlines, or edit-lines. During edit-mode, objects could also be translated while depressing the same mouse button when edit-lines were not selected. Unfortunately, it was difficult to avoid the edit-lines of a very small object. Consequently, users turned edit-mode off to translate a small object without inadvertently changing the size. A conflict resulted when the edit-mode status was common across workstations, and one designer was editing shapes while the other wanted to move small objects.

Another example of the need for independent environments involved points of view. When the designers were working on separate tasks, it was intuitive that they preferred to be in the view that best accommodated 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 was beneficial even when designers were working co-operatively on a single task. For example, when one designer was moving an object, the other could see the position of the object, which as shown from another view, was not apparent to the first designer from his view (see Fig. 3).

Takemura and Kishino [23] described an experimental task involving co-operative object manipulation in a three-dimensional, virtual reality environment where it was beneficial for the participants to perform the task from opposite views. Sarin and Greif [19] recognised the need for presenting the same data in different ways when participants have varied interests. Stefik *et al.* [24, 25] found that the broadcast of a participant's input command to modify an environmental parameter, such as the repositioning of a public window, would result in obstructing private work on other screens when independent environments were not allowed.



(a) **(b)**

Fig. 3. Misleading apparent object position due to point of view (a) non-aligned perspective view of first designer (b) aligned 'snap' view of second designer (note position of door in both views).

Although the participants were aware from the initial training session that neither points of view nor local cursor positions were shared, they did not always remember this. There were multiple occurrences of dialogues over the five sessions that resembled the following:

A: What were you planning to do with this (gestures with local cursor) block?
B: What block?
A: The one I'm pointing to.
B: I can't see your cursor.
A: The green one to the right of the door.
B: (Pause) Hmm. Where are you? What view are you in?
A: I'm looking down the blue axis.
B: Wait. (Switches views to match A) Oh. OK.

Levinson [26] pointed out that natural languages were intended for use in face-to-face interactions, and that this was particularly pronounced when the phenomenon of deixis, or indexical expressions, were present. Place-deictic words, such as here and there, and this and that, would sometimes require that appropriate gestures be transmitted between speaker and listener. In the situation described above, when transmission of the necessary type of gesture was not realized, participant A attempted to refer to the block by description: green and to the right of the door. This also proved unsuccessful because A and B did not share the same point of view, and this discrepancy in viewpoints was not immediately obvious to either. Additionally, had there been two green blocks, one to each side of the door as seen from A's view, it would not have been clear if the referent block was to the door's own left, or to the left from A's point of view. The mutual location of the referent object in the above dialogue was finally facilitated by using the reference frame provided by the *Teledesign* reference axes, which were fixed in the virtual world of the objects and contained three mutually orthogonal axes, each displayed in a different colour. The issue of points of view will be encountered again and discussed further in observations of experiment 2.

2.4.2. *Edit-access style*

Forced turn-taking was not found necessary for the two-person interactions that were observed during this set of tests. This was true for both types of tasks. The confusion caused by simultaneous editing was minimal. Occasionally, after some pause in spoken communication, two people would try to grab and move the same object without being aware of the other person's actions. During this experiment, the total time spent on recognizing and resolving such conflicts was on the order of five seconds per fifteen minutes. Stefik *et al.* [24] noted that when Colab participants co-ordinated their behaviour by using verbal cues, participants rarely changed the same data at the same time. They also found that social constraints could often be relied on for conflict resolution. Ellis *et al.* [21] also found that simultaneous editing was not chaotic due to the intervention of social protocol, and that, 'Collisions (were) surprisingly infrequent'. To further prevent accidental collisions, a 'busy signal' such as the graying of teleselected objects, or the use of a cloud-burst around changing text, has been implemented in text editors to warn other participants not to modify the same data [21, 24, 25]. Similarly, if the virtual positions of other designers in the meeting room were somehow represented in each workstation screen, accidental collisions would be less likely to occur.

The experiment participants were asked if they had a preference of edit-access mode for each of the two types of task. All of the participants 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. When they were asked if the sequential edit-access mode was more frustrating for one type of task than the other, there was no strong consensus. Some found that sequential access seemed less 'destructive' for the well-defined, inter-dependent task. Another participant commented that the room design tasks allowed for more creativity, and it was possible to use the waiting time during sequential access to think of things to do, whereas during the well-defined task, that participant knew what had to be done and thought that the task was more efficiently accomplished during the simultaneous access mode. Yet another participant flatly remarked that the sequential access mode was equally annoying during both types of tasks.

Ellis *et al.* [21] recognised that the sequential-access mode was 'limited to those situations in which a single active user fits the dynamics of the session, (and that it was) particularly ill-suited for sessions with high parallelism, inhibiting the free and natural flow of information'. Watabe *et al.* [27] reported that the type of floor-control mode selected depended 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, and the 'designation mode' was used when a clear chairperson was present. Brain-storming sessions used the 'free-mode' most often. Featuring the integration of inter-personal space and shared workspace, Clearboard [28] was designed for use by groups of two people. No floor-control mechanism was implemented for the drawing editor used in Clearboard. Sarin and Greif [19] also favoured 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'. Sarin and Greif [19] promoted 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'. It was also suspected that the simultaneous mode of edit access may become less appropriate as the size of the group increases.

3. Experiment 2: Software-usage instruction using Teledesign

3.1. *Experiment goals and design*

The first experiment had intended to study two types of interactions. It had been expected that the well-defined, interdependent task would entail a higher degree of collaboration than the loosely defined, loosely linked task, but we did not always find this to be the case. Some pairs of participants began by verbally dividing the tasks to be accomplished, and worked with minimal interaction until immediately before the end, where the disparate products of each person's efforts were hastily pieced together, even for the presumably interdependent tasks. We did not intend to criticize this manner of task accomplishment; instead we wanted to consider the possibility that another situation was needed to observe conditions marked by a higher degree of interaction.

It seemed promising that some type of instructional session, where a novice is paired with an expert, would provide opportunities for a high degree of interaction. Since new experiment participants would learn to use the *Teledesign* software, we decided to have experienced *Teledesign* users teach new participants how to use the software.

Several instances in which groupware was used in a teaching or training task has been documented. Ishii and Miyake [29] described the use of Teamworkstation for the remote teaching of calligraphy and of machine operation. While using Clearboard to teach and play backgammon, Ishii *et al.* [28] found differences in behaviour, specifically eye focus shifting between participants, during the teaching phase as compared with the rest of the session. The use of the TOPES system, an interactive graphics teleconferencing system used for the planning of building construction, to train new opeators to use the software, was described by Pferd *et al.* [12].

3.2. *Experiment task description*

The same equipment and set-up as reported for the first experiment were used. For the duration of the task, the participants were visually separated but could talk freely. First, the system expert stepped the novice through each of the available functions, demonstrating how to best take advantage of the system features. Then the team performed a simple collaborative design task as suggested by the novice, which lasted approximately half an hour. Examples of designs selected included those of a desk and chair set, a wheelchair, a trailer vehicle, and various room layouts.

3.3. *Observations*

3.3.1. *Pointing*

The need for a *telepointer*, a pointer that could be seen on multiple displays, was more distinct during the teaching phase than during either the rest of the session or experiment one. It was presumed that this was quite simply because the need to reference objects and parts of objects occurred more frequently during the instructional section. An example of a typical verbal exchange was: 'To change the shape of this block in this dimension, it would be easiest to select this corner, while editing from this view'. The multiple use of place-deictic words [26] within a single instruction amplified the need, demonstrated in experiment 1, for transmission of corresponding gestures. Regarding the remote teaching of machine operation on Teamworkstation, Ishii and Miyake [29] commented that direct pointing 'had greatly relieved (the instructors) of the extraneous burden of either verbalizing the locations of the buttons or pulling out the manual to indirectly point to the locations on the drawings'.

New users assumed that their local pointer was visible in the expert's display. Local pointer positions were not transmitted because sending the two-dimensional pointer location on the computer screen from one workstation to another would not be useful if the two workstations displayed the objects from different points of view. After the novice participants had been reminded that pointer positions were not transmitted, some participants improvised by selecting and 'shaking' the object of interest. They promptly discovered that this method had the disadvantage of disturbing the position of the referenced object, which in a few cases had involved some previous effort to locate correctly. A less intrusive method of gesturing involved selecting an appropriately dimensioned block to use as a pointer and moving the block to the vicinity of the object being discussed. Unless substantial effort was made to correctly position this pointer in three dimensions, this method was not a significant improvement over simply sending the two-dimensional local pointer position, measured relative to the computer screen, between workstations. Additionally, circumstances arose where the observer was in a point of view where other objects obstructed the view of the makeshift pointer.

3.3.2. *Point of view*

During the previous experiment, when a participant wanted to know the other designer's point of view, it was usually such that both designers could be in the same point of view to facilitate reference to objects or relationships between objects. During this experiment, knowledge of the other designer's point of view was often necessary for instructional purposes. For example, since the points of view between workstations were not automatically aligned, the expert did not always successfully demonstrate that certain tasks were more easily accomplished from certain points of view. The novice, not being able to see the experienced user change between views to perform particular tasks, tried to do everything from an non-aligned perspective view, only to become frustrated in the process. Furthermore, it was sometimes difficult for the expert to recognise that the student's difficulties were a direct result of editing in a view which was not best suited for the task.

3.4. *Possible solution*

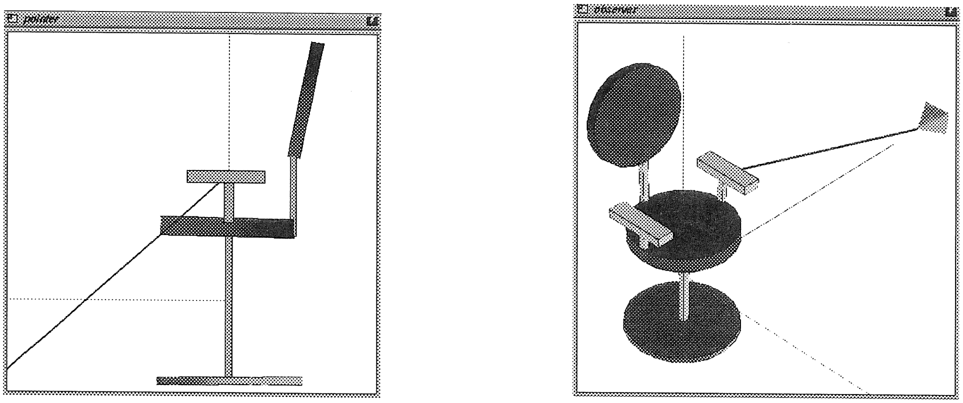
One conceivable solution would be to enforce WYSIWIS (What you see is what I see) during the teaching phase. Patterson *et al.* [30] suggested that in training simulations, instructors and students may want to have identical views to share both context and view. Lantz [16] also suggested the option of 'snapping back' to a common view. Stefik *et al.* [24, 25] 'teleported' people to view the appropriate windows. Enforcing strict WYSIWIS seemed to be a simple way of solving both problems: there would be no confusion about point of view, and pointing would be reduced to the two-dimensional case. In fact, strict WYSIWIS during an instructional segment, such that menus and cursor positions relative to menus are uniform across workstations, may be appropriate especially when there is a language barrier or other communication hindrance.

It was possible to an extent, to simulate WYSIWIS by having the instructors send their view to the student each time the point of view was changed. It was suspected that the pedagogical value of this method would be diminished if sudden, drastic changes in point of view that were not self-initiated became disorienting.

3.5. *Implementation of viewpoint*

To address these concerns, we chose to emphasize the ability to discern the other participant's point of view, by displaying a representation of the other designer's point of view at each workstation. The representation of a designer's point of view was embodied in a small pyramid comprised of colour-shaded polygons, positioned such that the tip of the pyramid aimed in the same direction as the designer's gaze. Descriptive of its primary function, this pyramid was called a viewpoint.

The viewpoint was also used to enable three-dimensional pointing that accommodated the disparate points of view of different designers. When pointing was desired, the pointer was activated using the pop-up menu. This resulted in the appearance of a line that extended from a lower corner to the centre of the screen. When an object was selected, the line brightened and terminated on the corresponding object. In the displays of the observing designers, a ray ran between the tip of the viewpoint representing the designer who was pointing and the selected object (see Fig. 4). Local pointers were coloured differently from remote pointers.



(a) (b)

Fig. 4. Implementation of viewpoint (a) view of designer who is pointing (b) view of designer who is observing.

4. Viewpoint: Directions for further development

The viewpoint represented a preliminary effort towards mutual awareness of fellow designers in the shared workspace. In its present form, the viewpoint effectively fulfilled the basic needs that we have identified for knowledge of other designers' points of view and the ability to point in three dimensions when designers had different points of view.

The most recent implementation of the *Teledesign* editor allowed points of view such that the designer may be located at any point on any sphere centred at the origin of the *Teledesign* reference frame, looking toward the origin of the reference axes. If this constraint were to be removed, such that it was possible to look from n y location in any direction, it would be necessary to provide orientation tools, such as indications of the designer's position relative to the reference axes and other designers.

It was likely that a three-dimensional input device, such as a spaceball or wand, would be more appropriate for position manipulation and pointing in a three-dimensional space. We believed that a spaceball would be appropriate for changing points of view, and rotating or possibly translating objects, while a wand would be suitable for three-dimensional pointing. The use of a data glove by Takemura and Kishino [23] to track a participant's hand position and shape in a shared virtual reality environment seemed to be an intelligent approach to combine object manipulation and gesture activities into one input device.

Further experiments to identify aspects that should be communicated between participants, such as gaze, activity and position awareness, would determine directions for development. It was not clear that a full reproduction of the human physical form would be the optimum representation of participants. We have inferred that approximating corresponding, familiar, physical situations would not always be the appropriate path to take. For example, Ishii *et al.* [28] found that communication of drawing activity between two people through a glass board, represented a situation that was more ideal for gaze awareness than the commonly practised meeting environments in front of white boards or over table tops.

5. Summary

Teledesign was built to investigate human-machine interface situations particular to co-operative work in three-dimensional computer-aided design. Experiments revealed that a simultaneous mode of edit access was preferred over a forced turn-taking mode of edit access for the observed two-person collab-orative efforts. 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 indicated the points of view of different designers and provided a method of pointing which was effective in an environment where arbitrary, contrasting points of views were allowed.

Acknowledgments

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

References

1. D. Clausing. Timed to Perfection. *Manufacturing Breakthrough*, **1** (1992) 35–40.
2. J. Martin. *Design of Man-Computer Dialogues* (Prentice-Hall, Englewood Cliffs, 1973).
3. C. V. Bullen and J. L. Bennett. 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, 1990) pp. 291–302.
4. A. Clement. Cooperative Support for Computer Work: A Social Perspective on the Empowering of End Users, in *Proceedings of the Conference on Computed-Supported Cooperative Work*, Los Angeles, CA, October 7–10, 1990 (ACM, New York, 1990) pp. 223–236.
5. J. Grudin. 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, 1988) pp. 85–93.
6. M. Mantei. 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, 1988) pp. 257–270.
7. S. Kiesler, J. Siegel and T. W. McGuire. Social Psychological Aspects of Computer-Mediated Communication, in *Computer-Supported Cooperative Work: A Book of Readings*, Ed. I. Greif (Morgan Kaufmann, San Mateo, 1988) pp. 657–682.
8. G. E. Kaiser, S. M. Kaplan and J. Micallef. Multiuser, Distributed Language-Based Environments, *IEEE Software,* **4** (1987) 58–67.
9. M. D. P. Leland, R. S. Fish and R. E. Kraut. Collaborative Document Production Using Quilt, in *Proceedings of the Conference on Computer-Supported Cooperative Work*, Portland, OR, September 26–29, 1988 (ACM, New York, 1988) pp. 206–215.

10. J. Galegher and R. E. Kraut. 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, 1990) pp. 65–78.
11. C. M. Neuwirth, D. S. Kaufer, R. Chandhok and J. H. Morris. Issues in the Design of Computer Support for Co-authorising and Commenting, in *Proceedings of the Conference on Computer-Supported Cooperative Work*, Los Angeles, CA, October 7–10, 1990 (ACM, New York, 1990) pp. 183–196.
12. W. Pferd, L. A. Peralta and F. X. Prendergast. Interactive Graphics Teleconferencing, *IEEE Computer*, **12** (1979) 62–72.
13. S. A. Bly. 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, 1988) pp. 250–256.
14. J. C. Tang and L. J. Leifer. 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.
15. S. Reder and R. G. Schwab. 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, 1990) pp. 303–316.
16. K. A. Lantz, An Experiment in Integrated Multimedia Conferencing, in *Proceedings of the Conference on Computer-Supported Cooperative Work*, Austin, TX, December 3–5, 1986 (ACM, New York, 1986) pp. 267–275.
17. D. J. Perreault. Interactive Computer Support for Remote Design Teams: A New Approach, MIT SM Thesis, Department of Electrical Engineering and Computer Science, Cambridge, MA, 1991.
18. T. Crowley, P. Milazzo, E. Baker, H. Forsdick and R. Tomlinson. 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, 1990) pp. 329–342.
19. S. Sarin and I. Greif. Computer-Based Real-Time Conferencing Systems, *IEEE Computer*, **18** (1985) 33–45.
20. E. M. Schooler, S. L. Casner and J. Postel. Multimedia Conferencing: Has It Come of Age? in *Proceedings of the 24th Hawaii International Conference on System Sciences*, 3 (ISI 1991), pp. 707–716.
21. C. A. Ellis, S. J. Gibbs and G. L. Rein. Groupware, Some Issues and Experiences. *Communications of the ACM*, **34** (1991) 38–58.
22. L. Shu. Groupware Experiences in Three-Dimensional Computer-Aided Design, MIT SM Thesis, Cambridge, MA, 1992.
23. H. Takemura and F. Kishino. Cooperative Work Environment Using Virtual Workspace, in *Proceedings of the Conference on Computer-Supported Cooperative Work*, Toronto, Canada, October 31–November 4, 1992 (ACM, New York, 1992) pp. 226–232.
24. M. Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning and L. Suchman. Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings, *Communications of the ACM*, **30** (1987) 32–47.
25. M. Stefik, D. G. Bobrow, G. Foster, S. Lanning and D. Tatar. WYSIWIS Revised: Early Experiences with Multiuser Interfaces, *ACM Transactions on Office Information Systems*, **5** (1987) 147–167.
26. S. C. Levinson, *Pragmatics* (Cambridge University Press, Cambridge, 1983).
27. K. Watabe, S. Sakata, K. Maeno, H. Fukuoka and T. Ohmori. 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.
28. H. Ishii, M. Kobayashi and J. Grudin. Integration of Inter-Personal Space and Shared Workspace: Clearboard Design and Experiments in *Proceedings of the Conference on Computer-Supported Cooperative Work*, Toronto, Canada, October 31–November 4, 1992 (ACM, New York) pp. 33–42.

29. H. Ishii and N. Miyake. Toward an Open Shared Workspace: Computer and Video Fusion Approach of Teamworkstation. *Communications of the ACM*, **34** (1991) 37–50.
30. J. F. Patterson, R. D. Hill and S. L. Rohall. 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.
31. J. Conklin and M. L. Begeman. 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.
32. L. Suchman. *Plans and Situated Actions* (Cambridge University Press, Cambridge, 1987).