Sharing Viewpoints in Collaborative Virtual Environments

Steven Valin, Andreea Francu, Helmuth Trefftz, and Ivan Marsic
Department of Electrical and Computer Engineering
Rutgers — The State University of New Jersey
Piscataway, NJ 08854-8058 USA
+1 732 445 0542
{valin, afrancu, trefftz, marsic}@ece.rutgers.edu

Abstract

In this paper we explore to what degree shared viewpoints in three-dimensional collaborative virtual environments enable effective collaboration. The paper applies research on shared viewpoints and telepointers to 3D environments. A lightweight Java-based tool for creating collaborative virtual environments was developed and used in the study. The system is realized as an application framework that can be customized to develop new applications. We conducted a series of experiments to assess the effectiveness of shared viewpoints on two simple tasks. Control groups were provided with telepointers. Experimental groups were provided with telepointers and shared views. The results indicate that for participants with access to both tools, shared views are preferred over telepointers for tasks involving joint exploration of either the environment or some object of common interest.

Keywords: Collaborative virtual environments, CSCW, groupware, viewpoint sharing.

1. Introduction

Collaborative virtual environments (CVEs) are increasingly being used for tasks such as military and industrial team training, collaborative design and engineering, and multiplayer games [15]. Many more applications are likely to emerge in the near future, given the availability and reduced cost of computers with powerful graphics boards and networking capabilities.

Much work in the area of enabling effective collaboration in CVEs has focused on developing the virtual reality metaphor to the point where it attempts to completely mimic collaboration in real environments [2],[3],[6]. In particular, much attention has been paid to user embodiment [1],[5],[16]. However, issues related to sophisticated user embodiments, such as facial expression and involuntary movement, require expensive virtual reality software and hardware. In addition, user embodiment and complete immersion in virtual worlds may not be necessary for a variety of collaborative tasks that can be performed in three-dimensional virtual environments. For instance, researchers have reported excellent results in enabling effective collaboration for performing such tasks as theatre set design [13] where participation in this collaboration was based upon a shared VRML model and did not require much more than a PC and a network connection. The system did have some shortcomings, including limited ability to modify the 3D model, and the lack of support for synchronous collaboration among multiple users.

While the current VRML standard does not contain any direct support for interaction among multiple users, recent work has focused on enhancements or extensions to VRML to support it. A common approach is to add a Java layer to enable multi-user collaboration.

Our motivation in developing cWorld was to support synchronous, multi-user construction of collaborative virtual environments and overcome the limitations of VRML and VRMLScript. We developed a graphical user interface for building 3D scenes using Java3D. We used DISCIPLE—a collaboration-enabling framework developed at Rutgers University—to enable multi-user, synchronous collaboration. The cWorld application is built as a JavaBean that is plugged into the DISCIPLE collaboration bus, and is thus made collaborative.

In developing cWorld, we are interested in understanding what minimum set of tools are necessary to enable effective synchronous, collaboration on simple tasks. It is well established that effective collaboration among multiple users relies heavily on their ability to refer to particular objects and to have other participants view those objects in a particular way [7],[9],[12]. Some of the same studies have also well documented the need for establishing a mutual orientation towards objects of common interest [7],[9]. In order to address issues associated with establishing mutual orientations, we added support for *shared viewpoints*—a strict form of WYSIWIS (What You See Is What I See) that allows one or more users to attach their viewpoints to another user's viewpoint and once joined, to share that viewpoint. It is a form of guided navigation where any of the users attached to the shared viewpoint may guide that viewpoint; i.e., not only do all users attached to a shared viewpoint see the same thing, but any of them may modify the shared viewpoint. Attachment to the shared viewpoint is a form of target-based navigation in that once a user has accepted an invitation to join a shared view, the user's viewpoint is immediately transformed to be the same as the viewpoint of the user that sent the invitation. Once a user detaches from a shared viewpoint, he or she is free to move about the virtual space using his or her own independent viewpoint. We also added support for *telepointers*. Telepointers in our system are implemented as 3D arrows that indicate the position and orientation of a user's viewpoint. They are used primarily to refer to objects in the shared virtual environment. In this paper, we describe the system we developed and the experiments we conducted in order to explore user preferences for shared collaborative viewpoints over independent viewpoints and telepointers. First the background on this issue is presented followed by a system overview. The technical part of the paper is followed by a study to evaluate the introduced concepts.

2. Background

Great success has been reported with collaborative theatre set design over the web [13]. In the 'Theatre in the Mill' study, collaborative theatre set design was achieved using a 3D VRML model of the Theatre in the Mill. Collaborative design was accomplished by passing stewardship of the model among the team members. In their paper, the authors refer to the IBM Theatre Project [10]—a system for immersive rehearsal in a virtual set. They point out that while it would be desirable to offer such an option, there are several reasons why they felt it inappropriate in their case. Among the reasons given were that immersive VR technology (i.e., headsets and body suits) is prohibitive for theatrical performances and far too expensive for most theatre groups. In addition, the authors point out that the 3D model was not designed to replace access to the actual space for activities such as rehearsal. Rather, it was designed to make sure that the limited time in the actual Mill theatre was used effectively (i.e., for rehearsal and performances rather than set design/redesign).

The authors of the Theatre of the Mill study reported that the use of the VRML model proved extremely valuable to traveling theatre companies. Set designers were able to view the performance space and try out ideas before committing to physical construction. Performers were able to familiarize themselves with the sets beforehand. However, the authors do point out shortcomings with the model. For instance, the relatively simple interactions supported by VRMLScript could not support complex operations, such as large-scale movement of lighting rigs and scenery redesign. Often these large-scale changes required a VRML developer to modify the model. Another shortcoming was that users had to take turns editing the model. There was no support for synchronous collaboration among multiple users.

Because the current VRML standard does not contain any direct support for interaction among multiple users, most VRML scenes run on a single machine and respond to a single user's input. Recent work has focused on enhancements or extensions to VRML in order to support multi-user, synchronous collaboration [4],[8],[14]. The basic approach is to add a Java layer to enable multi-user collaboration. However, this approach still suffers from the inherent limitations of VRML.

Motivated by the aforementioned successes, we wanted to develop a lightweight environment for web-based collaboration that would address the above limitations and still enable effective collaboration on certain tasks. Before attempting to implement a minimal system for supporting synchronous collaboration in 3D CVEs, we sought first to achieve an appreciation for the fundamental issues of multi-user collaboration.

WYSIWIS (What You See Is What I See) is a basic CSCW paradigm [17], which recognizes that efficient reference to common objects depends on a common view of the work at hand. Studies of workplace dynamics, media spaces, and more recently, collaborative virtual environments, have consistently demonstrated the need for participants to refer to particular objects and have other participants view these objects in a particular way while performing collaborative tasks [7],[9],[12]. Strict or nearly strict WYSIWIS is commonly found in two-dimensional collaborative applications such as shared whiteboard. However, even in a 2D world, strict WYSIWIS was found too limiting and relaxed versions were proposed to accommodate personalized screen layouts [17].

WYSIWIS makes less sense and is very uncommon in 3D virtual worlds. Collaborators need to navigate independently and accomplish their own goals, so they need independent views. However, this freedom brings also some impediments. Collaborators in media spaces can be frustrated by their inability to show each other artifacts such as paper or screen-based documents [12]. The Multiple Target Video (MTV) study showed that media spaces that simply provide multiple camera views were insufficient because multiple discontinuous views fragmented the workspace and prevented participants from establishing a mutual orientation towards artifacts involved in the collaborative task [7]. Many of the difficulties that participants experienced using the MTV system came from the need to switch between multiple, discontinuous views of remote spaces. The authors discovered that continuous movement allows us to change our focus of attention smoothly and thus enables us to interactively establish a mutual frame of reference, or mutual orientation, towards objects of interest.

A more recent investigation of object-focused interaction repeated basically the same experiments as the MTV study, but this time in a collaborative virtual environment (CVE) [9]. The study built on previous workplace and media space studies by examining the degree to which these issues were relevant in CVEs. The authors explored the extent to which their system provided participants with the ability to refer to and discuss features of the virtual environment. They found problems due to fragmented views of embodiments in relation to shared objects, caused in part by the limited

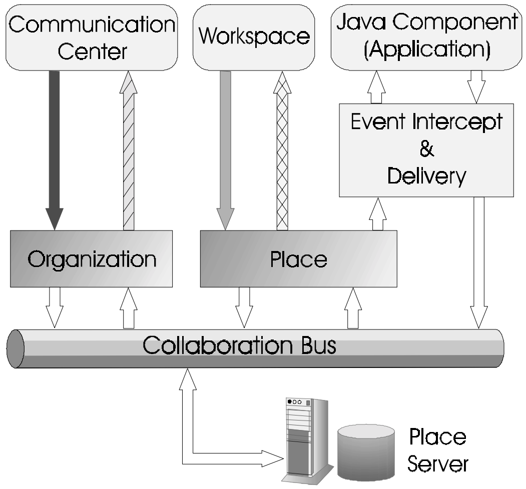


Figure 1: DISCIPLE architecture. Organizations and Places are abstractions implemented as multicast groups. They are represented in the user interface as Communication Center and Workspaces, respectively.

field of view (55^{o}) in the virtual environment. They also observed difficulties experienced by participants in understanding others' perspectives. Participants had great difficulty in understanding what others could see and expressed a desire for 'being in the other's position'. The authors proposed improved representations of others' actions and adoption of a form of target-based navigation providing users shortcuts for orienting towards targets.

In order to address the issue of 'being in the other's position', we propose the use of shared viewpoints—a form of guided navigation that allows one or more users to attach their viewpoints to another user's viewpoint. Once attached any participant may then transform that viewpoint. Thus, shared viewpoints provide a form of strict WYSIWIS in 3D CVEs, when needed. Attachment to the shared viewpoint is a form of target-based navigation as in [9]. When a user accepts an invitation to join a shared viewpoint, his/her own viewpoint is transformed to be the same as the viewpoint of the user that sent the invitation.

Sharing views in CVEs as a means to provide guided tours through virtual environments has been explored in [20]. The participants in the CVE are organized in a hierarchy of leaders and followers. Each participant can choose to follow a leader that guides the virtual exploration. If the follower does not manipulate his/her viewpoint, it is automatically attached to his/her leader's one. They also investigate how to reattach (non-abruptly) the follower's viewpoint to the leader's one once the follower finishes an independent wander. Our approach differs in several ways. The users in cWorld are not arranged in a hierarchy. Once several users agree to share viewpoints, anyone can take the lead. Also, once in a shared viewpoint everyone see exactly the same thing, while in [20], users are pulled along in the direction of the guide's movement.

In this paper we describe the system we have implemented and the experiments we have performed to assess user preference for single, shared viewpoints over multiple independent viewpoints when performing synchronous, collaborative tasks in a 3D virtual environment.

3. System Overview

Multi-user, synchronous collaboration is provided by the DISCIPLE framework. DISCIPLE is a mixture of client/server and peer-to-peer architecture. It is based on *replicated architecture* for groupware [19]. Each user runs a copy of the collaboration client, and each client contains a local copy of the applications (Java components) that are the foci of the collaboration. All copies of replicated applications are kept in synchrony and activities occurring on any one of them are reflected on the other copies.

Figure 1 shows the architecture of the DISCIPLE system. The set of participants is represented hierarchically as an Organization, and they meet in Places. DISCIPLE is organized in two independent layers: (1) the communication layer, called the *collaboration bus*, deals with real-time event exchange, dynamic joining and leaving, concurrency control and crash recovery; and (2) *the graphical user interface layer***,** which offers a standard user interface to every application bean imported into DISCIPLE. The collaboration bus comprises a set of communication channels, where the peers can subscribe to and publish information. In order to make the user aware of other users actions, the DISCIPLE GUI provides several types of group awareness widgets to all the imported beans. Telepointers are widgets that allow a given user to track remote users' cursors. In addition, the users can exchange messages, post small notes, and annotate regions of the bean window.

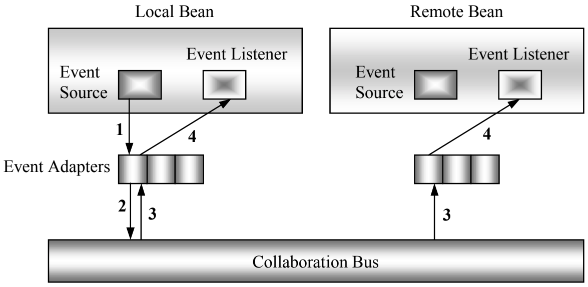


Figure 2: Event interception and symmetric distribution scheme in DISCIPLE: (1) The Event generated by the Event Source in the Local Bean, instead of being delivered directly to the local Event Listener, is intercepted by the associated Event Adapter and (2) sent to the Collaboration Bus. (3) The bus multicasts the event to all the shared Beans (remote and local). (4) Each Event Adapter receives the multicast event and delivers it to all listeners.

3.1. Sharing Java Beans

DISCIPLE is an *application framework*, i.e., a semi-complete application that can be customized to produce custom applications. The completion and customization is performed by end-users (conference participants) that at runtime select and import task-specific Java components—Beans and Applets. The DISCIPLE workspace is a *shared container* where Java Beans [18] can be loaded very much like Java Applets downloaded to a Web browser, with the addition of group sharing. Collaborators import Beans by drag-and-drop manipulation into the workspace. The imported Bean becomes a part of a multi-user application and all participants can interact with it. The application framework approach has advantages over the commonly used toolkit approaches in that with toolkit approaches the application designer makes decisions about the application functionality whereas in our approach the end user makes these decisions. We consider the latter better because it is closer to the reality of usage and the real needs of the task at hand.

According to the JavaBean event model, any object can declare itself as a source of certain types of events. A source has to either follow standard design patterns when giving names to the methods or use the Bean Information class to declare itself a source of certain events. The source should provide methods to register and remove listeners of the declared events. Whenever an event for which an object declared itself as a source is generated, the event is multicast to all the registered listeners. The source propagates the events to the listeners by invoking a method on the listeners and passing the corresponding event object.

Event adapters are needed since a collaboration module cannot know the methods for arbitrary events that an application programmer may come up with. Event adapters are equivalent to object proxies (stubs, skeletons), with the difference that the event adapters need to be registered as listeners of events so that the collaboration module is notified about the application's state changes. The process of event replication in DISCIPLE is illustrated in Figure 2.

A key feature of our framework is to make Beans collaborative without the need to alter their source code to adapt them to the framework. DISCIPLE loads the Bean and examines the manifest file in the Bean's JAR file for the information to automatically create the adapters. The adapters are generated with the code necessary to intercept the events, pass them to DISCIPLE to be multicast remotely and back locally, receive them after being multicast into the network, and pass them to the local bean. The code is then automatically compiled and the Bean's class path updated to contain the adapter classes.

3.2. cWorld Bean

The cWorld Java Bean enables synchronous, collaborative, multi-user building of collaborative virtual environments. It is built using the Java 2 SDK v.1.3.0 RC1 and the Java3D 1.2 Beta1 API OpenGL implementation. CWorld provides a graphical user interface for constructing and saving collaborative virtual environments. CWorld does not require any special hardware and can be operated using the keyboard and a mouse. It also supports the use of the Magellan SPACE Mouse [11]. This device provides a more natural six-degrees of freedom of movement for navigating the 3D space.

The software architecture of the cWorld bean is shown in Figure 3. The SPACE mouse manipulates either the viewpoint or graphics objects, depending on the selected mode. The Event Handler module intercepts user events and delivers the pertinent ones to the collaboration bus, which is registered as an event listener. Viewpoint events are delivered remotely only when view sharing is enabled. Multi-user collaboration is enabled by the DISCIPLE framework.

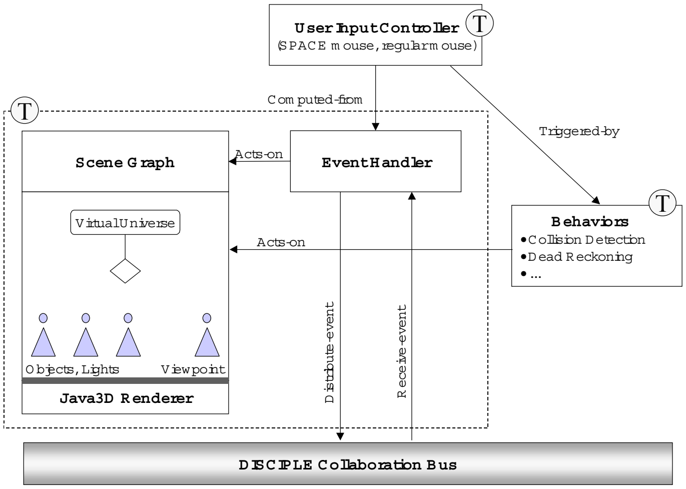


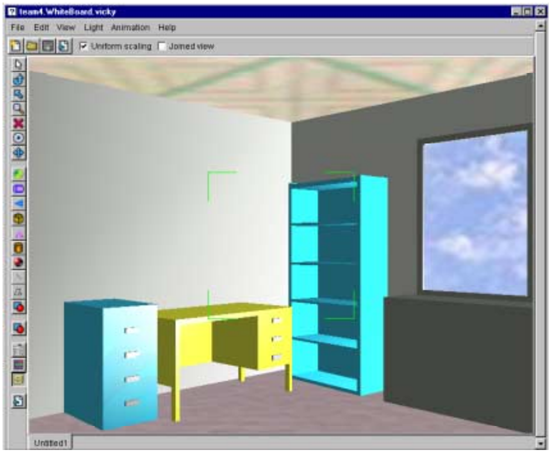
Figure 3: The architecture of cWorld. (T) symbolizes concurrent threads.

cWorld enables users to create new virtual worlds by providing 3D graphics editor functionality. Users may add primitive objects such as cubes, spheres, cones, as well as VRML objects. Once these objects are added to the scene, they may be transformed (translated, rotated, stretched, etc.). Once selected, the objects can be moved horizontally by displacing the sensor cap on the SPACE mouse. The user can also rotate object around its axis by rotating the cap on the SPACE mouse. This interaction proved to be very intuitive, and users learn it quickly. Through the use of a property editor, object properties such as color, shininess, highlight color, and texture mappings may be edited. CWorld also supports ambient lights, point lights, directional lights, and spotlights. Users may create complex objects by grouping simpler objects together. All objects can be made either public (i.e., globally accessible) or private (only the user that created them can access them). Additionally, any object may be fixed (position and properties) and thus becomes part of the background. A snapshot of a scene created using cWorld appears in Figure 4. Participants can alter their viewpoints by displacing and rotating the sensor cap on the SPACE mouse.

When a user opens a new or existing cWorld file, other users are invited to join in. At this point a collaborative session begins. Objects may be added, removed, or modified by the participants.

3.3. Viewpoints and 3D Telepointers

cWorld provides support for 3D telepointers (Figure 5) in addition to the 2D telepointers provided by DISCIPLE (which are not used in the tasks we describe). These devices function as a primitive avatar and appear when a user presses the appropriate mouse button. A 3D arrow is drawn at the position and orientation of the user's viewpoint. Telepointers are hidden by default and appear only while a user presses a specific button. The telepointers are a means for users to communicate to others where they are looking. Our implementation of telepointers is different from the *pointing arrows* in [6], in that those were drawn normal to the surface of the object of interest, while ours are drawn along the line of sight of the user. CWorld supports multiple, simultaneous telepointers. Multiple users may simultaneously invoke the use of telepointers and have all of the other participants in the collaboration view their telepointers (assuming it is in their field of view).The cWorld bean also supports the use of shared, collaborative viewpoints.



☑ Uniform scaling □ Joined view

Figure 4: A sample CVE built using cWorld. Note: objects must be placed within crosshairs in order to be selected.

When a user joins a cWorld session, he/she is provided with his/her own, independent view of the world. However, at any time a user may wish to share his or her particular view of the virtual space with others. Alternatively, users may wish to view the space as someone else sees it. This is accomplished using shared views. A user may invite others to join in a shared view. Users indicate their desire to join in the shared view by selecting this option from the menu bar. Once in a shared view, all users view the world from the viewpoint of the user that sent the invitation. Furthermore, once users have joined in a shared view, any of them may rotate or translate that view. Once a user chooses to leave the shared view, the user is returned to their own independent viewpoint.Methodology

3.4. Hypothesis Tested

In this experiment we wanted to investigate how users might use shared views and the degree to which use of shared views helps or hinders collaboration on two simple tasks.

3.5. Subjects

The 27 subjects ranged in age from 18 to 32 and had varying levels of experience with computers and video games. Five subjects had never played video games and ten had very little experience with video games. Eleven subjects had moderate experience with video games (between one and five hours per week). Only one subject reported playing video games for more than five hours per week. All participants indicated they were comfortable using a computer and mouse, but only three had previous experience with 3D collaborative virtual environments. Potential participants were asked to form their own groups of three before registering to participate. They were not further re-assigned to form more or less experienced teams.

3.6. Procedure

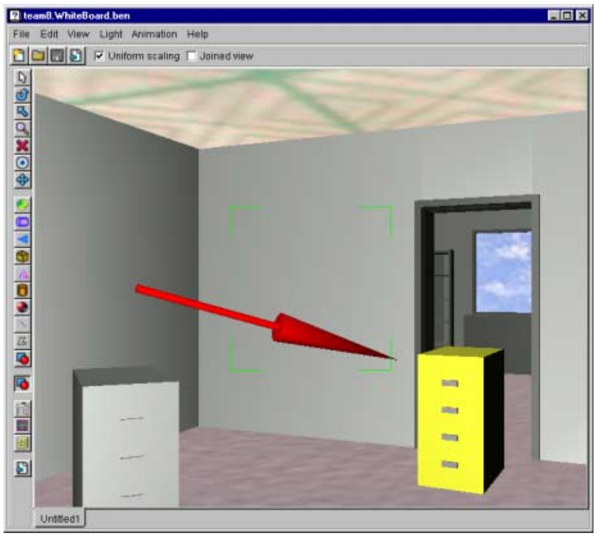
The experiment was comprised of three tasks performed by teams of three subjects at a time. There were nine teams in total. The teams were divided into two groups: four control groups and five experimental groups. The control groups performed the tasks using only telepointers and independent viewpoints. The experimental groups were given the additional option to use shared views. Each team was seated in the same office. They were placed in different cubicles so they could not see each other but could hear each other. Participants used Windows NT workstations connected via an Ethernet LAN. Workstations were equipped with both a normal PC mouse and a Magellan SPACE mouse device (Figure 6).

Using cWorld, we built two virtual environments and the furniture objects used in the experiment. All of the furniture objects were public. Participants' own furniture appeared blue to them, while it appeared gray to others. Also, once a participant selected a furniture object it appeared yellow to them until they deselected the object or selected another. Object and viewpoint movement was disabled in the *y*-axis in order to prevent 'flying'.

3.6.1. Task 1. The Room Orientation Task

The primary purpose of this task was to familiarize participants with the Magellan SPACE Mouse and the cWorld interfaces. The task is as follows:

1. Each subject is seated at a workstation where a cWorld session has been started.



team8.WhiteBoard.ben

☑ Uniform scaling □ Joined view

Figure 5: A three-dimensional telepointer example.

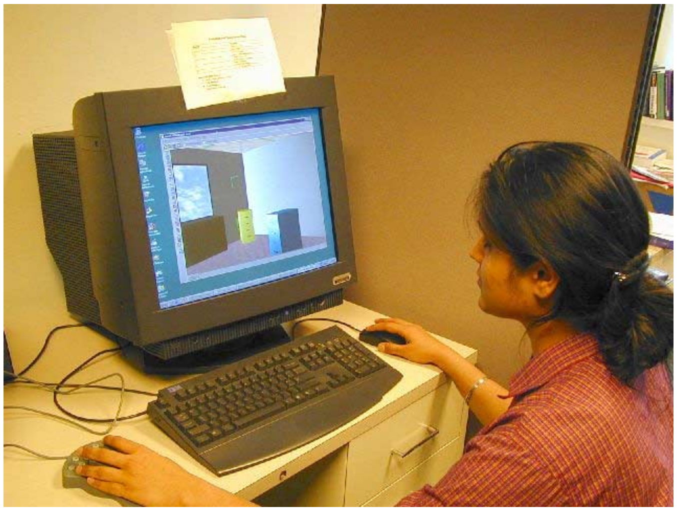


Figure 6: A participant in a collaborative session. Note: participants' workstations had navigation and object manipulation hints on top of the screen.

2. A research team member instructs participants in the use of cWorld and the Magellan SPACE Mouse. This training includes moving in the environment, adding and moving objects, using telepointers, and using shared views (experimental group only).
3. Next, the researcher instructs each participant to place a furniture object at a particular location. After all participants have placed their object, they are instructed to each take turns indicating to the other participants, which object they placed using the telepointers and shared views (experimental group only).

3.6.2. Task 2. The Room Design Task

This task was designed to evaluate the degree to which shared viewpoints may enable effective collaboration in a 3D environment. Three participants enter a cWorld space that contains an empty (virtual) office. Each participant is instructed to imagine that they will all be moving into a shared office. They each have a desk, a cabinet, and a bookcase that they wish to move with them. They are instructed to use cWorld as a tool to decide where they would like to have the moving company place their furniture when it is moved to their new office. Each participant is given their own set of (virtual) office furniture that they are asked to place in the room however they wish, without breaking certain rules; e.g., furniture cannot block doors or windows, desks may not be stacked on top of one another, etc. The task was made more difficult by the fact the furniture fits into the room in only a limited number of configurations. Thus, in order to accomplish the task, all users must participate (they have their own furniture to place) and all users must collaborate (since it is unlikely that all of the furniture will fit into the room on the first try). There is also a competitive component in task 2: Users should want to place their own furniture in prime locations (e.g., next to the window or away form the door) and they may want to finish first.

3.6.3. Task 3. The "What's Wrong with this Room?" Task

The purpose of Task 3 was to compare the results of task 2 with a task that appeared to be more collaborative in nature and less competitive. The task is as follows: Participants are placed in a cWorld environment that contains two rooms separated by a doorway. The two rooms are almost identical except for some minor differences in the way the furniture was placed. One room is designated the model room and the other is designated the working room. Participants are asked to identify and correct the differences in the working room so that it exactly resembled the model room. In order to insure that the participants collaborated (and do not just immediately correct the imperfections that they themselves only saw), we instruct them to get agreement from the other subjects before making any changes to the working room.

We evaluate the effectiveness of shared views by recording the following:

1. The amount of time required to complete the task.
2. The time spent in shared view. (Experimental group only).
3. The number of times the users joined their views.
4. And, through the use of pre- and post-experiment questionnaires.

The pre-experiment questionnaire included questions about the subjects' background, such as experience with video games and input devices. Post-experiment questions were designed to evaluate participants' subjective impressions about the level of team collaboration and the effectiveness of the cWorld interface in supporting collaboration.

3.7. Results

The results are summarized in Tables 1 and 2 below. The control group took on average 533 seconds to accomplish task 2 (σ = 166). The experimental group took on average 586 seconds to accomplish task 3 (σ = 169).

On task 3, the control group took on average 525 seconds (σ = 153). The experimental group took on average 429 seconds (σ = 146) to accomplish task 3.

78% (21/27) of all participants believed that their team had collaborated will on the tasks. 80% (12/15) of the experimental group participants believed that their team had collaborated well on the task while 75% (9/12) of control groups believed the same.

On task 2 experimental groups infrequently used shared views and spent an average of 3% of their time in shared views.

On task 3 experimental groups moved in and out of shared views and spent an average of 8% of their time in shared views.

Table 1: Summary of results on task 2.

	Average time to complete task (seconds)	σ	Believed their team collaborated well on task	Found Telepointers helpful in completing task	Among those that believed their team collaborated well on task, the number that found Telepointers helpful
Control Group	533	166	73% (11/15)	60% (9/15)	73% (8/11)
Experimental Group	586	169	80% (12/15)	53% (8/15)	58% (7/12)
			Relative time spent in Shared Viewpoints	Found Shared Viewpoints helpful in completing task	Among those that believed their team collaborated well on task, the number that found Shared Viewpoints helpful
			3%	53% (8/15)	50% (7/12)

Among experimental group participants that felt their team had collaborated well on task 2, over half (58%) felt that shared views helped them in accomplishing the task.

Among experimental group participants that felt their team had collaborated well on task 3, a clear majority (67%) felt that shared views helped them in accomplishing the task.

Table 2: Summary of results on task 3.

	Average time to complete task (seconds)	σ	Believed their team collaborated well on task	Found Telepointers helpful in completing task	Among those that believed their team collaborated well on task, the number that found Telepointers helpful
Control Group	525	153	80% (12/15)	53% (8/15)	58% (7/12)
Experimental Group	429	146	87% (13/15)	40% (6/15)	54% (7/13)
			Relative time spent in Shared Viewpoints	Found Shared Viewpoints helpful in completing task	Among those that believed their team collaborated well on task, the number that found Shared Viewpoints helpful

		8%	73% (11/15)	77% (10/13)

On task 3, we observed that participants used the shared views more often. This is perhaps due to the fact that they did not have parallel, independent tasks to perform, but rather were working jointly to identify the differences with the working room. The following dialog is representative of participant interaction when using shared views:

RAFAEL:

I would like to show you one of the changes I think we should make…

Do you want to join views?

CECILIA:

Yes.

PAHOLA:

Hold on… OK.

RAFAEL [*now manipulating the shared view*]:

I think this bookcase has to be moved to the other side of the window. Do you agree?

CECILIA:

Yes, that's exactly what I was thinking.

PAHOLA:

OK. Sounds good. Who wants to move it?

RAFAEL:

Let me do it…

We also observed that participants used the shared views as a means target-based navigational shortcut. For instance, in task 3, one group used shared views as a means to be transported between the two rooms:

VICKY: [*in the working room*]

Say again which object should be closer to the window…?

ADAM: [*in the model room*]

Let's join views and you'll see what I mean.

VICKY:

OK

[*Adam invites Vicky to join views*]

[*Vicky accepts Adam's invitation and is immediately transported to Adam's viewpoint*]

I see… I'll go back and move the file cabinet.

[*Vicky presses button 5 on the SPACE mouse and navigates back to the working room*].

Table 3 contains selected participant responses to the question of whether or not they found shared views helpful.

Table 3: Selected participants' comments on shared views.

1	"Yes, because you can share information and allow an easier communication with your team."
2	"Yes, because it saves time."
3	"Yes, they are helpful because it is useful to know other people's point of view."
4	"It is useful because it allows one user to show others exactly what they want to through their own eyes."
5	"Did not use it. It was too slow."
6	"No, because we found that we could verbally communicate our intentions."
7	"Not for these particular tasks, though I think shared views may be necessary for other applications using cWorld."

For all subjects (experimental as well as control groups) that felt they had collaborated well on task 2, 67% felt that telepointers helped. When we consider only experimental group subjects (i.e., those that also had access to shared views) only 53% found telepointers useful in accomplishing task 2.

For all subjects (experimental as well as control groups) that felt they had collaborated on task 3, 52% felt that telepointers helped. When we consider only experimental group subjects (i.e., those that also had access to shared views) only 40% found telepointers useful in accomplishing task 3.

There were also some unexpected uses of telepointers. For instance, one participant stated that telepointers were a nice way to indicate one's location to other team members.

Table 4 contains selected participant responses to the question of whether or not they found telepointers helpful.

Table 4: Selected participants' comments on telepointers.

1	"In Task #2 it definitely was helpful."
2	"Telepointers is a nice way for others to know your present location."
3	"Point to space where we put file cabinets."
4	"Permanent mini-telepointers would be nice to show where all the other members are looking."
5	"In task #2, we wanted to put the filing cabinets in one corner, and we used the telepointer to determine which corner."
6	"I used the telepointer in task 3 to see if the rest of the team liked the position of the filing cabinet."
7	"I think we did not use it because we use the shared view, that in certain way could replace the telepointer."
8	"Since we could talk, there was no need for them."
9	"If not using shared views, telepointers made it easy to show others what I am looking at or talking to them about."
10	"I found telepointers unintuitive. Again, these may be useful for other applications."
11	"They served no purpose that could not be solved with verbal communication."
12	"I pointed at the file cabinet that I had placed."
13	"But they did not work well. When I held down button 5, the pointer flickered at best, and my teammates did not see it well."
14	"No, we forgot to use them."
15	"I forgot they were available."

In Table 4, the participant that provided comment 13 was pressing the wrong button—he should have used button 4 to activate the telepointer. Participants that provided the last two comments used shared views.

4. Discussion

The data collected on average task completion times shows that on average the control groups outperformed the experimental groups on task 2, while the experimental groups outperformed the control groups on task 3. However, the large variances associated with these times, render the data inconclusive. These large variances may be a result of:

- Participants' widely varying previous exposure to video games. Those with some video game experience appear to have done better at performing the tasks and making use of the tools provided to them.
- The nature of the tasks was not appropriately tailored to the use of sharing viewpoints; i.e., telepointers may have been equally effective for the tasks we defined.

Given the fact that we did not form the participant groups based on their previous experience with video games and that the participants' experience varied widely, this was probably the greatest factor responsible for the large variances in task completion times. In addition, potential participants were asked to form their own groups. This led to teams of participants where they all had roughly the same amount of experience on video games: from not at all to very experienced.

The fact that participants made greater use of shared viewpoints in task 3 would seem to indicate that the usefulness of shared views is task-dependent. Therefore, it is reasonable to assume that there may be tasks that would more fully exploit shared views. From our observations of when shared views were used, we conclude that shared views provide greater benefit on tasks that are either instructional in nature or in which joint exploration of either the environment or some object of interest is necessary.

Another approach would have been to also assess the quality of the tasks performed. However, we opted not to do so for the following reasons:

- Even though most participants took great care in aligning the furniture, they did not appear to be motivated to compete for prime office space locations.
- It was inherently difficult to assess the quality. Minor differences in the layout of the furniture are hard to appreciate. Instead we decided to give participants a set of rules to follow and used the time it took to accomplish the task as a means of assessing the quality of the collaborative effort.
- Quality, in a way, was embedded in the measurement of the time to complete the task.

Based on our observations of the participants and their responses to the questionnaire, users found both telepointers and shared views useful. However, they found shared views more useful on task 3, than on task 2. On task 2, 58% of participants that felt they had collaborated well, found shared views helpful. On task 3, the number was 67%. In addition, among those users that had a choice on using telepointers or shared views on task 3, they clearly preferred shared views. On task 3, 67% of users found shared views helpful, where those users had access to both tools and believed they had collaborated well. For telepointers, only 42% found them helpful.

We also observed that among those that did not find shared viewpoints helpful, the overwhelming majority had little or no experience with 3D environments or video games. It would appear that prior experience on video games plays a decisive role in determining participants' effective use of the tools we provided, and ultimately, their ability to accomplish the tasks quickly and efficiently. The more experience they had with video games, the more they made use of the tools and found them to be helpful. This leads us to conclude that we should either avoided naïve participants or provided greater training in the use of the tools.

We also confirmed previous results reported by others that users attempt to use verbal communication as a means to overcome limitations associated with making their intentions known. Comment 6 in Table 3 and comment 11 in Table 4 illustrate this point.

Many participants stated that they would have liked to have a greater level of knowledge of where others were in relation to themselves. This is illustrated by comments 2 and 4 in Table 4. This suggests that even for the simplest tasks performed in synchronous, collaborative environments there may be a need for peripheral monitoring of co-collaborators. While there were numerous suggestions on how to provide this peripheral monitoring (including two-dimensional maps and radar screens), only one participant explicitly mentioned avatars.

On a related note, our current implementation of attaching to another's view does not provide a smooth transition. However, the discontinuity associated with attaching and detaching from shared viewpoints did not appear to significantly hinder the effectiveness of shared views. This was probably due to the fact that users were collaborating in very simple and small virtual environments where they could quickly develop a mental image of the space. In more complex environments this discontinuity would cause greater difficulties, as would the lack of user embodiment.

5. Summary

Three dimensions on the desktop remain most individuals' experience of virtual environments. It is important to return to this context whenever discussing production of publicly available collaborative tools. The purpose of this study was to explore under what circumstances sharing viewpoints is sufficient for enabling effective collaboration. The goal was to design a lightweight, web-based tool without the need for elaborate embodiments and sophisticated virtual reality equipment. Furthermore, we wanted to investigate in what situations sharing viewpoints would be more or less effective than using telepointers.

We found that sharing viewpoints did enable effective collaboration and is more effective than telepointers for some tasks. At the same time, we found that participants in collaborative 3D virtual environments desire at least some form of peripheral monitoring of co-collaborators.

We also found that Java3D and the DISCIPLE framework provided an easy-to-use, scalable, efficient means for enabling synchronous, multi-user collaboration in three-dimensional collaborative virtual environments.

Our continuing work involves adding support in cWorld for simple avatars. Users will be able to create their own avatars using the cWorld toolset, and then have their avatar attached to their viewing platform. Our future experiments will explore whether it is necessary to provide pseudo-humanoid avatars, or whether something as simple as a hand or a pointed-finger may suffice. We are also investigating the use of 2D maps and radar views for supporting peripheral awareness of co-collaborator activities. Finally, we are currently adding support for smooth attachment to and detachment from shared viewpoints.

The DISCIPLE project source code, sample beans, and documentation are freely available at:

http://www.caip.rutgers.edu/disciple/

Acknowledgments

A. Wanchoo, A. Krebs, B. Dorohonceanu, and K. R. Pericherla contributed significantly to the software implementation. The research reported here is supported in part by DARPA Contract No. N66001-96-C-8510, NSF KDI Contract No. IIS-98-72995 and by the Rutgers Center for Advanced Information Processing (CAIP).

References

[1] S. Benford, J. Bowers, L. E. Fahlen, C. Greenhalgh, and D. Snowdon, "User embodiment in collaborative virtual environments," *CHI '95 Proceedings*, ACM Press, pp.242-249, 1995.

[2] S. Benford and C. Greenhalgh, "MASSIVE: A collaborative virtual environment for teleconferencing," *ACM Transactions on Computer-Human Interaction*, 2(3):239-261, September 1995.

[3] T. K. Capin, I. S. Pandzic, D. Thalmann, and N. M. Thalmann, "Realistic avatars and autonomous virtual humans in VLNET networked virtual environments," *Virtual Worlds on the Internet*, J. Vince and R. Earnshaw, eds., IEEE Computer Society, Los Alamitos, pp.157-173, 1998.

[4] J. Carson and A. Clark, "Multicast shared virtual worlds using VRML97," *Proc. of the 4th Symposium on the Virtual Reality Modeling Language (VRML'99)*, Paderborn, Germany, pp. 133-140, 1999.

[5] T. Era, K. Kauppinen, A. Kivimäki, and M. Robinson, "Producing identity in collaborative virtual environments," *Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST'98)*, Taipei, Taiwan, pp. 35-42, November 1998.

[6] E. Frécon and A. A. Nöu, “Building distributed virtual environments to support collaborative work,” *Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST’98)*, Taipei, Taiwan, pp.105-113, November 1998.

[7] W. Gaver, A. Sellen, C. Heath, and P. Luff, “One is not enough: Multiple views in a media space,” *Proc. of INTERCHI ’93*, ACM, New York, pp.335-341, April 1993.

[8] T. Goddard and V. S. Sunderam, “ToolSpace: Web based 3D collaboration,” *Proc. of the 4th Symposium on the Virtual Reality Modeling Language (VRML’99)*, Paderborn, Germany, pp.161-165, 1999.

[9] J. Hindmarsh, M. Fraser, C. Heath, S. Benford, and C. Greenhalagh, “Fragmented interaction: Establishing mutual orientation in virtual environments,” *Proc. of the ACM 1998 Conference on Computer-Supported Cooperative Work (CSCW'98)*, Seattle, WA, pp.217-226, November 1998.

[10] IBM Theatre Projects, http://www.ibm.com/sfasp/theatre.htm

[11] LogiCad3D GmbH, Magellan/SPACE Mouse, http://www.logicad3d.com

[12] P. Luff, C. Heath, and D. Greatbatch, “Tasks-in-interaction: Paper and screen based doumentation in collaborative activity,” *Proc. of the ACM 1992 Conference on Computer-Supported Cooperative Work (CSCW'92)*, Toronto, Canada, pp. 163-170, 1992.

[13] I. J. Palmer and C. M. Reeve, “Collaborative theatre set design across networks,” In *Virtual Worlds on the Internet*, J. Vince and R. Earnshaw, eds., IEEE Computer Society, Los Alamitos, pp.253-261, 1998.

[14] K. Saar, “VIRTUS: A Collaborative Multi-User Platform,” *Proc. of the 4th Symposium on the Virtual Reality Modeling Language (VRML’99)*, Paderborn, Germany, pp.141-152, 1999.

[15] S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*. Addison Wesley, New York, 1999.

[16] D. Snowdon and J. Tromp, “Virtual body language: Providing appropriate user interfaces in collaborative virtual environments,” *Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST’97)*, pp.37-44, 1997.

[17] M. Stefik, D. G. Bobrow, S. Lanning, and D. Tatar, “WYSIWIS revised: Early experiences with multiuser interfaces,” *ACM Transactions on Information Systems*, 5(2):147-167, April 1987.

[18] Sun Microsystems, Inc. JavaBeans API Specification, http://java.sun.com/beans/

[19] W. Wang, B. Dorohonceanu, and I. Marsic, “Design of the DISCIPLE synchronous collaboration framework,” *Proc. of the 3rd IASTED International Conference on Internet, Multimedia Systems and Applications*, Nassau, The Bahamas, pp.316-324, October 1999.

[20] E. Wernert and A. Hanson, “A framework for assisted exploration with collaboration,” *Proc. of IEEE Visualization’99*, San Francisco, October 1999.