Adding Immersion To Collaborative Tools

Edward Swing[1]

National Security Agency

ABSTRACT

While three dimensional collaborative environments have been used for industrial design or interactive games, workgroup collaboration has largely remained in the two-dimensional realm. In this paper, we examine the collaborative capabilities of the Collaborative Virtual Workspace, and how it is used. We then describe our effort to augment this system with an immersive display. By developing an immersive interface to an existing collaboration tool rather than adding collaboration to an immersive world, we hope to discover the advantages and pitfalls that immersive collaboration environments might offer. The first step in this process is to design the immersive environment to provide natural interactions for the activities users normally perform while working within the collaborative environment.

CR Keywords and Subject Descriptors: C.2.4 [Computer Communications Networks]: Distributed Applications; I.3.2 [Computer Graphics]: Distributed/Network Graphics; I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques;

Additional Keywords: VRML, Collaborative Virtual Environment, Computer Supported Collaborative Work.

1 INTRODUCTION

Many virtual environments which are both immersive and collaborative are designed for a specific purpose, such as games or collaborative design [9]. Others are initially designed as an immersive world primarily, with the capability to collaborate added to them [3, 6, 10]. With this project, we start with an existing collaborative environment, and augment it with an immersive interface. Our goal is to observe the benefits derived from adding the immersion to a collaborative environment which is already actively used by a variety of users, and compare how the users' interactions within the immersive environment differ from their interactions within the traditional environment,

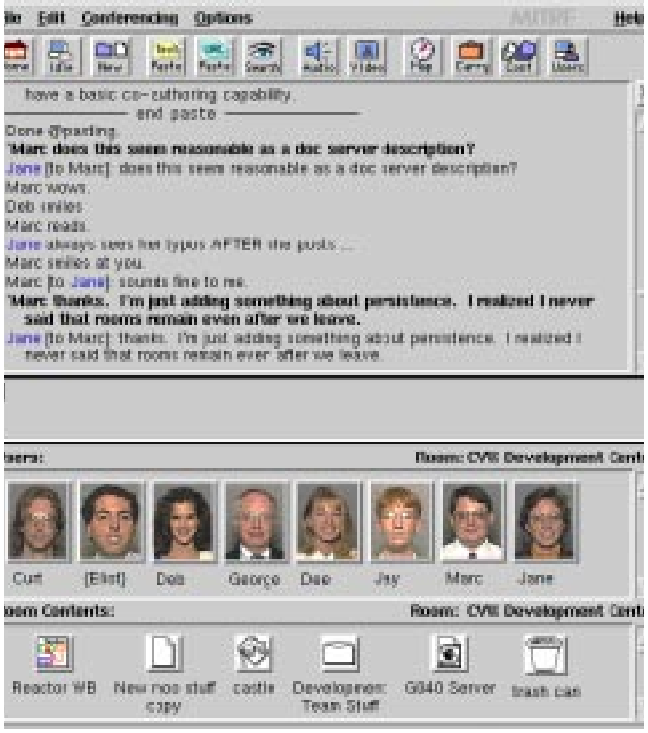
Like many large corporations or other large government agencies, the National Security Agency is investigating collaborative technologies for internal use. The Collaborative Virtual Workspace (CVW) has been gaining acceptance within NSA because of its flexibility and power. Because CVW is used by a body of users at NSA on a daily basis, it provides a useful testbed of actual collaborative activity. By using CVW as a starting point, we will be able to observe how providing an immersive display might benefit users.

The development and testing of the immersive interface is an ongoing process. The first step is designing the immersive setting to provide natural analogs of common activities users normally perform while working within the collaborative environment.We begin with an introduction to the standard CVW client, and follow with an explanation of the rationale and implementation for the 3D interface. This is followed by a discussion of user reaction to the initial fielding of the augmented system.

2 OVERVIEW OF CVW

The Collaborative Virtual Workspace (CVW) is a prototype collaborative computing environment created by the MITRE Corporation [6]. CVW maintains a persistent virtual space where users can communicate, collaborate, or share documents. Audio, video or simple text-based interaction are all supported within CVW.

The CVW collaborative environment is divided into virtual buildings, each of which has several floors and a number of rooms on each floor. Typically, the building might be devoted to a high-level organization or general topic, while each floor is devoted to a specific organization, or more specific topic. The individual rooms host the actual collaborative sessions.



have a basic co-authoring capability.
——— end paste ———
Done @pasting.
'Marc does this seem reasonable as a doc server description?
Jane [to Marc]: does this seem reasonable as a doc server description?
Marc wows.
Deb smiles.
Marc reads.
Jane always sees her typos AFTER she posts ...
Marc smiles at you.
Marc [to Jane]: sounds fine to me.
'Marc thanks. I'm just adding something about persistence. I realized I never said that rooms remain even after we leave.
Jane [to Marc]: thanks. I'm just adding something about persistence. I realized I never said that rooms remain ever after we leave.

Figure 1: Sample CVW Screen

CVW supports a number of different type of objects. In addition to the users themselves, CVW provides folders, notes, whiteboards, URLs, and other documents. These can be imported from external programs and be shared with other users. CVW allows multiple

[1] Attn: E03/CWEF, 9800 Savage Rd., Ft Meade, MD, USA. Phone: 301-688-3336; email: edswing@ncsc.mil

users to collaborate on a shared document or whiteboard, or restrict access to objects they create.

Users can speak or emote publicly within a room, or whisper to another user privately. They can page other users, or locate them within the entire CVW environment. Users can also place a proxy of themselves in a room, and monitor activity in one room while remaining in another.

All public interactions occurring within a room are displayed in a textual scrollback, identifying other users and their actions within the room. Rooms also hold the shared documents, which persist within the space. Users can restrict access to rooms as well, locking out others during private conferences. Rooms may also have a recorder to act as a note-taker.

2.1 Internals of CVW

The interface that users interact with is known as the CVW client, shown in Figure 1. This communicates with the CVW server through socket connections. The CVW server coordinates the individual users within a virtual building, keeping track of locations of users and other objects within the space. The server also provides for the dialog between users.

CVW uses enhanced MUD (Multi-User Dungeon), Object-Oriented (MOO) technology for its messages and controls. MUDs have widely been used on the internet for interactive role-playing games; recently developers have explored applying this same technology for other applications [7]. CVW uses LambdaMOO, developed by Xerox PARC, as a foundation. Many extensions and capabilities have been added to LambdaMOO to create CVW, such as audio and video conferencing and the shared whiteboard.

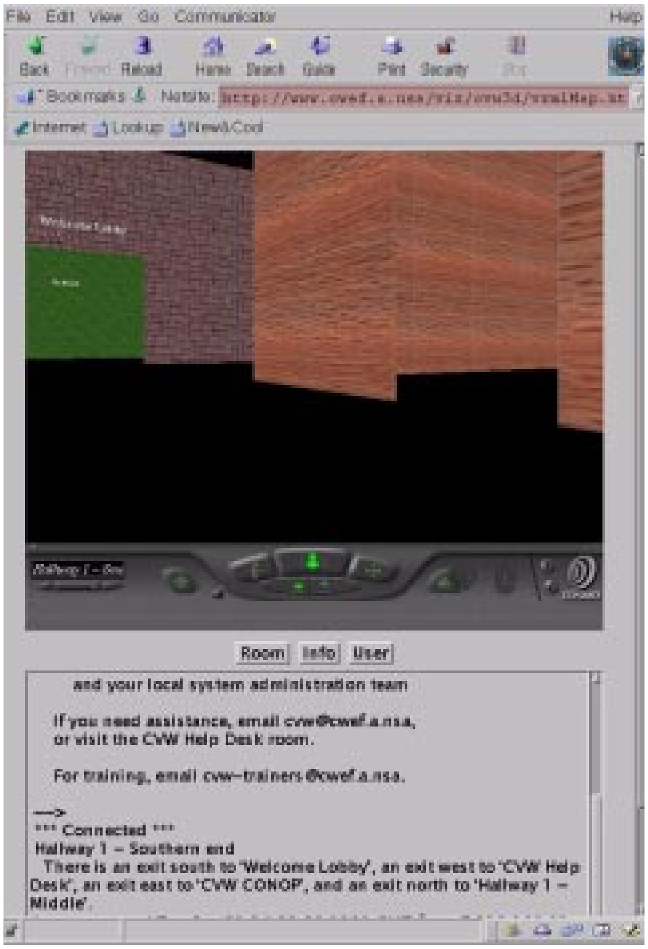
CVW also uses IP/Multicast technology to provide the underlying network infrastructure for multipoint video and audio conferencing. Each room in CVW has a unique multicast address on the NSA intranet in which the audio and video conferences are held.

Documents are managed through a document server, which not only provides a central location for the document, but also provides version controls for group editing. The document exchange interaction is based on the HTTP protocol. As users import documents into the CVW environment or open documents for editing or viewing, the document server provides the documents in the same fashion as a web-page. Document types are identified by MIME types, thus allowing users to add particular media types easily into a room. CVW will use MIME to identify the proper application to launch for editing a particular document.

CVW is freely available, and is being developed under the Open Source model by the MITRE Corporation [4].

3 AUGMENTING THE CVW CLIENT

The CVW immersive client (also known as CVW3D) is being developed to explore how immersion helps or hinders a collaborative experience. Users are able to enter the virtual room immersively, and interact with other users through avatars. Documents and other objects are represented in the space as well, and interactions with other users and objects are provided in an intuitive fashion.


File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Netsite: http://www.cwef.a.nsa/vrml/cvw3d/vrmlMap.ht
Internet Lookup New&Cool
Welcome Lobby
Room Info User

and your local system administration team

If you need assistance, email cvw@cwef.a.nsa,
or visit the CVW Help Desk room.

For training, email cvw-trainers@cwef.a.nsa.

--->
*** Connected ***
Hallway 1 – Southern end
There is an exit south to 'Welcome Lobby', an exit west to 'CVW Help Desk', an exit east to 'CVW CONOP', and an exit north to 'Hallway 1 – Middle'.

Figure 2: CVW3D Immersive Client

The immersive display uses VRML (Virtual Reality Modeling Language), and therefore runs within a web browser, as shown in Figure 2. The room dialog is maintained in a panel below the immersive window, so the room scrollback feature is maintained. During certain operations, the room dialog is overlaid by an alternate display. For instance, when querying and privately sending messages to another user, a user panel is displayed, which shows that other user's identity.

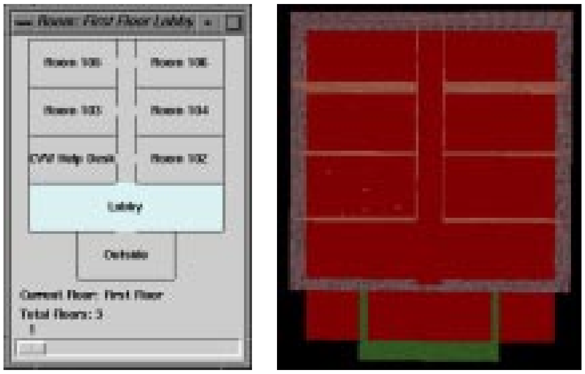
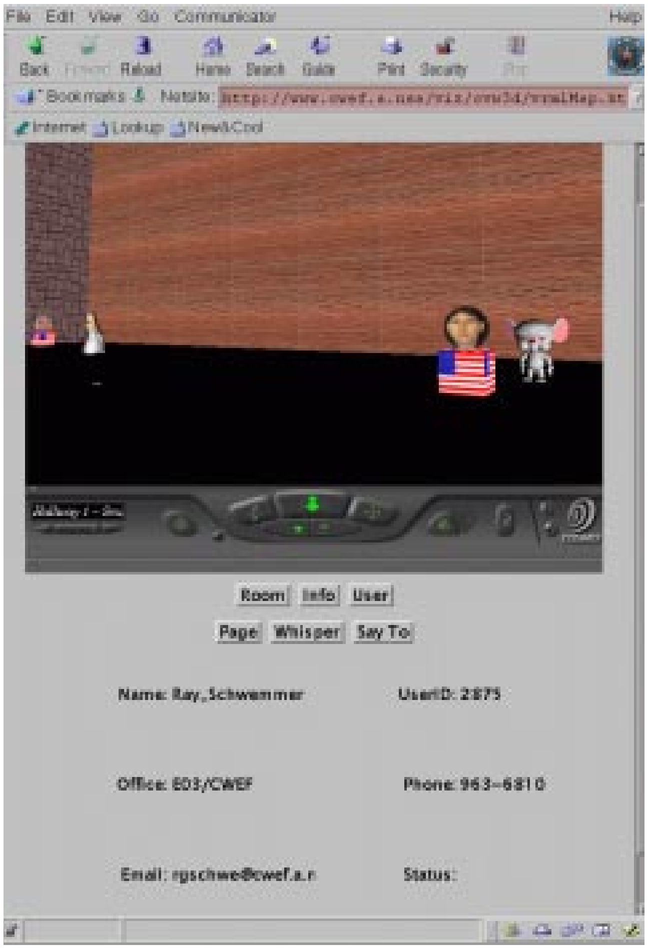


Figure 3: CVW Floor Layout: Original (left) and Immersive (right)

One floor is represented at a time within CVW3D. The floor replicates the design used within the CVW client, so the room geometry is identical (see Figure 3). The user may enter or leave rooms using the navigation controls, or by using preset VRML viewpoints to jump into any room on the current floor. Like the standard CVW client, CVW3D only updates the current room's contents dynamically, and displays room contents only when a user moves into a room. Some other collaborative immersive environments also restrict updates to a specific region to reduce the updates necessary for the scene [2, 11], however the approaches are quite varied.

3.1 Users and Avatars

Every other user is represented by an avatar within the virtual room. The default avatar is a simple model, showing a user's face on a sphere, and affiliation on a cube (shown as US flags in Figure 4). However, CVW3D can incorporate more detailed models or full motion avatars, allowing the user to customize his personal presence.


File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Netsite: http://www.cwef.a.usa/mis/cvw3d/vrmlMap.ht
Internet Lookup New&Cool
Room Info User
Page Whisper Say To

Name: Ray,,Schwemmer UserID: 2875

Office: E03/CWEF Phone: 963-6810

Email: rgschwe@cwef.a.r Status:

Figure 4: Sample Avatars

Most other multi-user immersive environments focus on maintaining a consistent representation to all participants. This includes coordinating avatar movement, so that if a user moves across a room, his avatar representation walks. Coordinating this type of behavior among all participants typically requires a great deal of communications, and numerous strategies have been developed to handle the necessary coordination [12]. Many systems also coordinate object ownership and manipulation, but this is typically for moving the objects within the scene.

Unlike other immersive environments, the avatars' physical locations within the CVW3D room are randomly generated, and do not have any particular meaning. Since the CVW server and the traditional CVW clients do not maintain a user's physical position within a room, dynamically updating the avatar's position is not feasible at this time. Similarly, avatars are currently static - they do not gesture within the scene. Future developments may allow avatars to gravitate toward one another as they converse, or use preset gestures, such as waving, to accompany certain user activities. This may require additional functionality within the CVW server.

All avatars are clickable, so the user can display information about the room's other inhabitants easily. The additional information includes their phone number or organization, and is displayed within a panel of the user interface. Within the user interface, the CVW3D client provides a mechanism to communicate directly to another user, either emoting, speaking, or whispering privately. This mimics the functionality of the traditional CVW client.

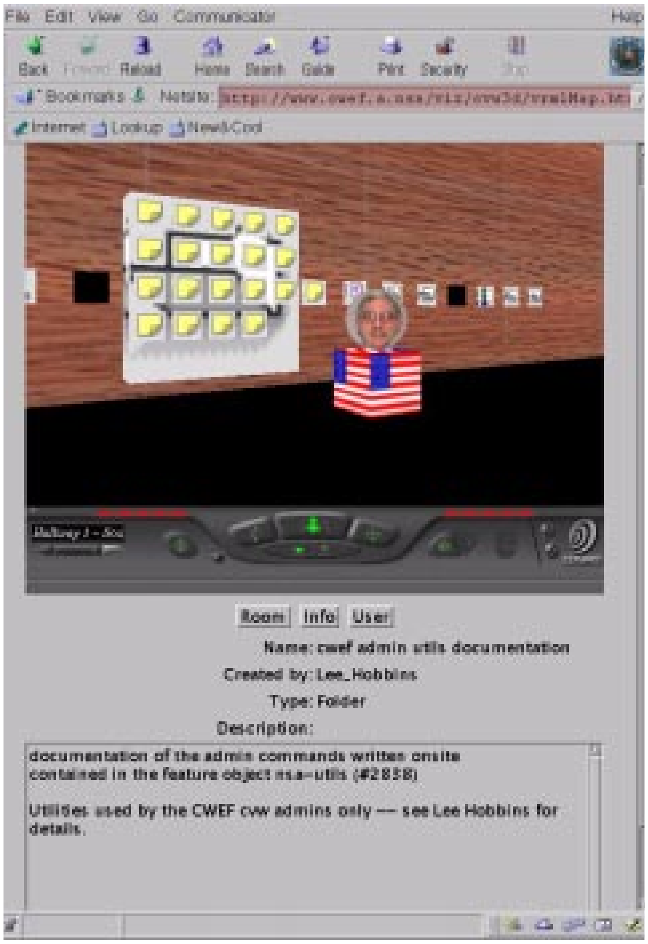
3.2 Documents and other Room Objects

Besides users and their avatars, rooms can also contain various other objects, such as documents, notes, whiteboards, and web references. Within the standard CVW display, a room object is represented by an icon. In the immersive display, these icons are mapped onto tiles, which are then placed at eye level along one wall of the room, simulating a bulletin board (see Figure 5).

A user within the virtual space can select an object from the wall, and query its attributes, move it into his carrying folder, or open it for display. Since the documents from the CVW document server are handled using MIME types, it is simple to display documents from the CVW3D client in the same fashion. Users can query an object for additional information, such as owner, permissions, and last modification time. Likewise, when a user wishes to open an object for display, the web-browser opens the object as if it were a web-reference.

While the full CVW client handles checking documents out and updating them, this feature is not yet available within the CVW3D client. For the same reason, it does not yet handle the interactive Whiteboards.

Certain object types are handled differently. Notes, which are simple text, are displayed within the information panel. Folders are a collection of documents within a room; they can contain any type of object. A user opening a folder within a standard CVW interface would see a separate window displaying the contents of that folder. In the CVW3D immersive interface, the folder icon itself expands and displays the contents as smaller items on the folder. In this fashion, the user can interact with folder contents identically with other objects within the immersive room.


File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Netsite: http://www.cwef.s.osa/vis/cvw3d/vrmlMap.ht
Internet Lookup New&Cool
Room Info User

Name: cwef admin utils documentation

Created by: Lee_Hobbins

Type: Folder

Description:

documentation of the admin commands written onsite
contained in the feature object nsa-utils (#2858)

Utilities used by the CWEF cvw admins only — see Lee Hobbins for details.

Figure 5: Documents along Wall, with Open Folder

3.3 Navigating

Users moving around the immersive world can use the standard VRML browser navigational controls to move around the environment. Preset views are also set for each room, so a user can jump from one room to another very quickly.

A user can change floors by using the virtual elevator within the lobby of each floor (see Figure 6). This will place the user on a different level within the virtual environment. In reality, the room geometry does not change, since all floors have identical layouts. However, the room contents and labels are reset to match the new floor.

Certain rooms within the CVW environment maintain access controls, allowing only certain users into the room. In the traditional CVW client, a user who tries to enter is not allowed, and the CVW server will refuse to transmit any information about the barred room aside from its name. The immersive client erects a physical barrier (a door) over the doorway when a user tries to enter a barred room. Even if a user manages to enter a room by manipulating his browser controls, the information is still hidden because it is controlled by the CVW server.

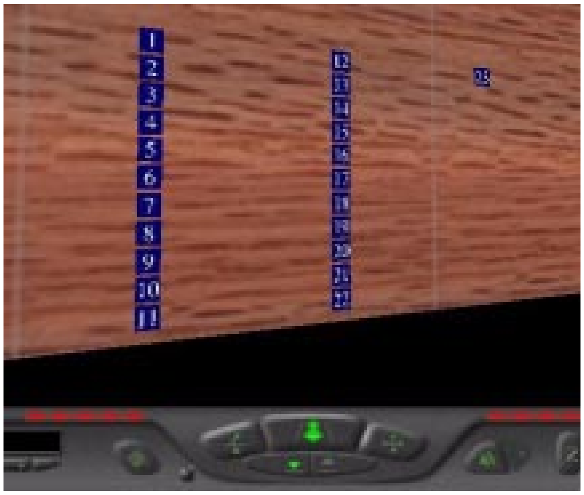


Figure 6: Elevator and VRML Browser controls

3.4 Software Architecture

Because the CVW system is being used in operational environments, the initial development focussed on creating an immersive environment which provided many of the same features and functionality. We wanted to provide the users with the familiar virtual buildings and rooms of CVW, but provide a new method of interacting with them. Enhancing the interface to provide unique capabilities offered by the new immersive experience will be the second phase of development.

CVW uses a server for each building as the centralized coordinator for that building. When entering the environment, CVW users can select the building they wish to enter from several different buildings in the campus. The CVW server was preserved without modification, so interaction with objects within CVW only differs at the client side where users interact with the objects and each other.

The CVW client has several different versions currently in operation. A Tcl/Tk version is available, but development on this version has stopped. A Java client is now provided publicly, with the same functions and features of the earlier Tcl/Tk client.

The CVW3D client is developed using a modified version of the Java source code, using Java 1.1. The display is created using VRML (Virtual Reality Modeling Language), which is controlled using the EAI (External Authoring Interface). The CVW3D client runs as an Applet within a web browser.

The floor is loaded from a VRML file, and then the textual labels for the rooms are applied whenever a new floor is entered. At the same time, the viewpoint labels are updated to show the new room names. Users and objects within the rooms are added dynamically whenever the user enters or leaves a room. Whenever a user enters a room, the CVW server provides all contents to the CVW client being used; when the CVW3D client receives this information, it loads the avatars for all other users present, and generates the geometry for each object in the room.

The CVW3D client maintains awareness of the user's movement from room to room with VRML Proximity Sensors. Each sensor corresponds to a single room; the VRML browser sends the events corresponding to entering or leaving rooms to the Java EAI, where that information is then forwarded to the CVW server.

Avatars within the scene can be simple models or H-Anim compliant humanoids [1]. The users will have the ability to select their avatar. A directory of avatars (one per user) is kept on the web server; a default avatar is used if there is no avatar for a specific user.

Because the CVW server is in use for operational tasks, the developmental web server used to serve up the CVW3D client ran on a separate web server. Because applets do not allow socket communications with any address except the web server providing the applet, it was necessary to create a temporary relay server on the web server to relay the messages between the CVW server and the CVW3D client applet. This approach avoided the cumbersome Java Security authentication, and the small increase in the network traffic did not adversely affect the web server.

As the code for the immersive client developed, the original Java source code was logically separated into several modules. One of these modules handles communications with the CVW server. The second module handles the various objects within the CVW environment, maintaining their identity and properties which are used by the client. A third handles the user interface. By providing a well-defined separation of the various modules, creating additional alternate user interfaces should be straightforward.

4 USER FEEDBACK

Currently the CVW3D client is still in development, and initial user testing has begun. The interface has generated a great deal of interest from both users and other developers, who are interested in experimenting within the immersive environment. Obviously, enhancing the CVW3D client to implement the full functionality of the traditional client is necessary for a true evaluation; this task is underway.

The initial user reaction has been favorable, but with reservations. Some users expressed concern over the interface, which is quite different from the standard CVW client. Many users were unfamiliar with three-dimensional interfaces, but were able to use the VRML browser after a few minutes of instruction. Once they became familiar with the navigation and other controls, they were much more relaxed. Other comments focused on specific changes and enhancements to the interface.

Several users have commented on the locations of the avatars within the immersive room. Since the CVW server does not maintain location information of objects within the room, the locations of the avatars are randomly generated. Several users have suggested creating arrangements, such as placing the avatars around a virtual conference table, or moving avatars towards each other if they are talking with one another. Another asked about hearing conversations in the adjoining rooms if he stood in the doorway between rooms. Obviously providing a meaning to the avatars' locations could provide important contextual information that would certainly augment the user experience.

One concern voiced by some users is the potential field-of-view problem. Users within the immersive environment may rely primarily on the visual display to distinguish who is in a room. If another user is behind them, the user might not be aware of their presence. This can cause problems if the user says something to the room that he or she does not want the other user to hear. We hope that by preserving the room scrollback capability, a user can keep aware of other inhabitants of the room, but more testing is needed. We added a feature to display the room scrollback whenever a new room is entered, based upon a user's suggestion.

Enhancing the virtual experience, such as moving avatars within the room or providing gestures, may enhance the immersive experience without requiring additional communication with the CVW server. Enhancements of this type may be added initially as programmed actions - for instance, waving at a new user entering the room. Users suggested several other capabilities, such as referring to documents by pointing with their avatar, which would require some enhancement to the CVW server capabilities.

When asked about potential uses for the immersive interface, we received a number of comments. VRML models can be included as a type of document within a room, and then a user opening this document could display a 3D model within the room. One user suggested a 3D whiteboard enhancement, allowing users to mark up such a model. Another suggestion involved display of flight scenarios or similar events, which would take advantage of the 3D functionality; users are already displaying such scenarios, usually by marking up images interactively within a whiteboard.

Users suggested other potential uses of the interface as an augmentation to their current CVW experience. One suggested allowing users to interactively arrange documents in the rooms, allowing a user to group the documents into several virtual bulletin boards. Another suggested displaying video on one wall, similar to a movie screen. User preference was divided between including everything in one window or using additional popup windows for media (as the traditional CVW client does). Since users can move around in relation to the documents, those preferring to add everything to the immersive display found that approaching documents to view the contents was quite intuitive.

5 FUTURE WORK

MITRE developers have been excited about the prospect of providing the CVW3D client to their customers. They have provided a number of suggestions for enhancements within the virtual environment. The CVW3D interface will be provided to the Open Source code base once the initial development has been completed.

In addition to exploring user-defined arrangements, we hope to explore display strategies for organizing the documents and other objects within the room. Like the avatars, there are a number of possible methods to arrange the documents within each room, including document clustering, hierarchies and other methods. Such techniques may enhance the collaborative experience.

Other future developments include adapting the interface to use Java3D as an alternate immersive client. Since the source code for the user interface is logically separated from the remaining source code, only modification of a limited number of routines will be necessary. This viewer-independent approach has also been used by other collaborative environments [8].

We hope to provide the CVW3D interface to the user community as a whole. CVW has a strong user base within NSA, and should provide an excellent testing ground. We have already seen fascinating results of how novice users interact with a virtual environment which is familiar to them, and this exploration continues as we enhance the interface. By continuing to explore how immersion augments a user's collaborative experience, we will understand how to improve such spaces.

References

1. M. Beitler, "HANIM 1.1 Compliant VRML 97 Models", http://www.cis.upenn.edu/~beitler/hanim/.

2. W. Broll, "Populating the Internet: Supporting Multiple Users and Shared Applications with VRML", Proceedings of the Second Symposium on the Virtual Reality Modeling Language, pp. 33-40, ACM, 1997.

3. J. Carson and A. Clark, "Multicast Shared Virtual Worlds using VRML97", Proceedings of the Fourth Symposium on the Virtual Reality Modeling Language, pp. 133-140, ACM, 1999.

4. Collaborative Virtual Workspace Open Source Web Site, http://cvw,mitre.org, MITRE Corporation

5. J. Davidson and L. Deus, "A Case Study in Technology Transfer of Collaboration Tools", The Edge (http://www.mitre.org/pubs/edge), July 1998, MITRE Corporation.

6. P. Diefenbach et al, "Building OpenWorlds", Proceedings of the Third Symposium on the Virtual Reality Modeling Language, pp. 33-38, ACM, 1998.

7. R. Evard, "Collaborative Networked Communication: MUDs as Systems Tools", Proceedings of the Seventh Systems Administration Conference (LISA VII), pp. 1-8, Monterey CA, 1993

8. T. Goddard and V.S. Sunderham, "ToolSpace: Wed Based 3D Collaboration", Proceedings of the Fourth Symposium on the Virtual Reality Modeling Language, pp. 161-165, ACM, 1999.

9. W. Lamotte et al, "Visinet: Collaborative Visualization and VR over ATM Networks", IEEE Computer Graphics and Applications, Vol. 17, No. 2, Mar. 1997, pp. 66-75

10. R. Lea et al, "Sony Community Place: Architecture and Performance", Proceedings of the Second Symposium on the Virtual Reality Modeling Language, pp. 33-38, ACM, 1997.

11. K. Saar, "VIRTUS: A Collaborative Multi-User Platform", Proceedings of the Fourth Symposium on the Virtual Reality Modeling Language, pp. 141-152, ACM, 1999.

12. M. Wray and V. Belrose, "Avatars in Living Space", Proceedings of the Fourth Symposium on the Virtual Reality Modeling Language, pp. 13-19, ACM, 1999.