REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Feb 95	

4. TITLE AND SUBTITLE
A Formative Evaluation of CU-See Me

5. FUNDING NUMBERS

6. AUTHOR(S)
Michael Bibeau

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFIT Students Attending:
Virginia Polytechnic Institute

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/CI/CIA
95-007

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DEPARTNEMT OF THE AIR FORCE
AFIT/CI
2950 P STREET, BDLG 125
WRIGHT-PATTERSON AFB OH 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release IAW AFR 190-1
Distribution Unlimited
BRIAN D. GAUTHIER, MSgt, USAF
Chief Administration

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS

15. NUMBER OF PAGES
140

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

A FORMATIVE EVALUATION OF CU-SEEME

by

Michael J. Bibeau

Roger W. Ehrich, Chairman

Computer Science

(ABSTRACT)

CU-SeeMe is a video conferencing software package that was designed and programmed at Cornell University. The program works with the TCP/IP network protocol and allows two or more parties to conduct a real-time video conference with full audio support. In this paper we evaluate CU-SeeMe through the process of Formative Evaluation. [3] [9] [16] [24] We first perform a Critical Review of the software using a subset of the Smith and Mosier Guidelines for Human-Computer Interaction. [23] Next, we empirically review the software interface through a series of benchmark tests [3] that are derived directly from a set of scenarios. The scenarios attempt to model real world situations that might be encountered by an individual in the target user class. Designing benchmark tasks becomes a natural and straightforward process when they are derived from the scenario set. Empirical measures are taken for each task, including completion times and error counts. These measures are accompanied by critical incident analysis [2] [7] [13] which serves to identify problems with the interface and the cognitive roots of those problems. The critical incidents reported by participants are accompanied by explanations of *what* caused the problem and *why*. This helps in the process of formulating solutions for observed usability problems. All the testing results are combined in the Appendix in an illustrated partial redesign of the CU-SeeMe interface.

A Formative Evaluation of CU-SeeMe

By

Michael Bibeau

Thesis submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
Computer Science

APPROVED:

Roger W. Ehrich, Chairman

Robert C. Williges

Mary Beth Rosson

February 20, 1995

Acknowledgements

I would like to personally thank all the individuals who helped me with this project. Rob Mohn was a tireless volunteer during the testing process as a conference participant and deserves due recognition. I would also like to thank Dr. Ehrich for his support with advice and equipment, Dr. Williges for the use of the HCI lab and his many inputs into this document, and Dr. Rosson for her suggestions and time.

Finally, I would to thank Norri and Forrest for understanding when I was at the lab late at night.

I dedicate this work to my son, Forrest.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. Introduction

1.1 Motivation

The growing popularity of the Internet in our schools has brought to the classroom a large number of client software programs traditionally geared toward the intermediate to expert computer user. Eager to exploit the potential power of the Internet, many schools have started experimenting with the use of Internet-based video-conferencing systems such as CU-SeeMe from Cornell University.[1] Currently, CU-SeeMe is free to all "self-selected beta testers" which has greatly enhanced its popularity in both the public and private sectors. CU-SeeMe allows users on Internet-based computers to conduct audio-supported video-conferences with one or several other parties. There is currently much debate over the amount of network resources used by CU-SeeMe and how to improve its treatment of these resources. This has motivated developers to concentrate their efforts on the underlying technical aspects of the software in order to increase performance while decreasing bandwidth usage. These efforts have certainly brought about several improvements in the software performance. However, even though CU-SeeMe is gaining popularity with the non-technical users, much of the interface design has retained a more technically oriented look than seems appropriate. During a demonstration to a group of local teachers associated with the Blacksburg Electronic Village (BEV) project, demonstrators noticed the audience struggling with the software and suggested that what is needed is a Formative Evaluation [9] of the current CU-SeeMe interface design. A Formative Evaluation of the interface at this point in the development can serve two purposes. First, it will give the developers valuable input into the quality of the current design from the users' viewpoint. Second, it will uncover

[1] From information gathered on the KIDSPHERE newsgroup, va-pen.mailing-list.kidsnet, and archived messages in the askERIC searchable gopher.

various approaches to solving usability problems associated with the design. Finally, it will provide insight into potential improvements in functionality for the CU-SeeMe system.

The most popular use of CU-SeeMe today seems to be recreational and curious exploration. Some groups conduct activities like "The virtual Coffee House," a place where people conduct a multi-party conference and take turns reading poetry. Recently, "The House of Blues" broadcast a concert featuring Stevie Wonder through Internet lines with the use of CU-SeeMe. There are even adult-oriented sites that allow people to meet via live video. The U.S Department of Education has broadcast its "town meetings"[2] using CU-SeeMe, and NASA has its own TV station called NASA Select TV, continuously broadcasting over the Internet via CU-SeeMe. With this kind of interest in the possibilities of video conferencing growing daily, the developers of CU-SeeMe must work to improve the CU-SeeMe interface so that it will remain a popular and effective tool even when "the novelty wears out". The best way to start improving is with a Formative Evaluation using participants that represent the future predominant user-class.

1.2 Internet Software

A Formative Evaluation of CU-SeeMe is necessary to help improve the usability of the CU-SeeMe software package among the increasing number of novice users [19] that are gaining access to the Internet. With the growth-rate of the Internet increasing daily, there is a rising need to simplify the use of the Internet and the software that lives there. The novice Internet user finds himself plagued by countless buzzwords, and acronyms causing the Internet to become a fairly cryptic place to visit. Although there have been a number of software packages evolving recently that remove some of the mystery from the net, the novice still faces terms like FTP, TCP/IP, IP Address, URL, kbps, and other network-specific jargon. Novices often find themselves having great difficulty achieving even the most basic tasks on the network, like reading news groups.

---

[2] E-mail message forwarded from CU-SeeMe Listserver
Originator, Jane Smith <jds@kudzu.cnidr.org> Tues, 11 Oct 94.

Much of the current software still requires a basic understanding of the underlying technical aspects of the Internet. The growing popularity of the Internet and the increasing usefulness of Internet resources requires that future software products become better grounded in the foundations of Human-Computer-Interaction(HCI) Guidelines. The needs of current users are such that client software must focus on the task at hand, for example, the intent to send an e-mail message, and de-emphasize the process or underlying protocols involved with the task, such as the proper construction of an e-mail header.

1.3 Education and the Internet

Since Internet access became available to many public school systems, school teachers have used the Internet to increase their effectiveness as educators. [15] [17] [18] Using the Internet, some teachers communicate and collaborate with other teachers throughout the world, greatly expanding their professional abilities. Other teachers use the Internet in their classrooms to allow students to collaborate with, or compete against students from other parts of the country, or even the world. The possibilities are limited only by the teachers' imagination and the available resources. Considering the potential power of the Internet as a tool for education, the research community needs to explore each and every possibility that will make the technology more accessible and useful to everyone in the field . . . on both sides of the desk. CU-SeeMe is one such tool that brings some of the usefulness of the Internet into the classroom.

1.4 Current Trends

Video-conferencing is one of the more exciting areas of Internet use and has evolved rapidly in recent times. By using the existing network structure, digital video, and inexpensive software, it is possible to have a real-time video and audio conversation with individuals anywhere in the world. Video-conferencing on a personal computer is starting to break the $2,000 price barrier but much of the computer-based video-conferencing technology in place today is still relatively expensive

(some complete systems as high as $10,000) and based on proprietary protocols. Although the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) has developed video-conferencing standard H.320, without a *ratified* protocol standard the only way for anyone to make video-conferencing widely available is to make it inexpensive and flexible. [14] If both ends of a conference need the same software package, then it must be accessible to both systems and both budgets. CU-SeeMe from Cornell University is now making it possible for a wide range of individuals to communicate through this technology. With relatively inexpensive hardware, the CU-SeeMe software package and a standard Internet connection, any individual can become an active part of this new kind of virtual community. CU-SeeMe includes real-time video and audio, the ability to conduct multi-party video-conferences and widely distributed broadcasts.

1.5 Goals of the Formative Evaluation

The potential benefits of a video-conferencing system for educational use are certainly notable. Teachers can expand their resources and their students can become active voices in the world instead of just within their classroom walls. By making video-conferencing readily available to the masses with systems such as CU-SeeMe, we can explore the technology in a real-world setting. Through this Formative Evaluation of CU-SeeMe, and the input of testing participants (both teachers and learners), we can identify some important issues in the design of a user interface for a video-conferencing system such as CU-SeeMe. We can critically evaluate the existing interface structure and build new ideas for not only improving the existing interface, but also for expanding its functionality to meet the potential needs of future users.

2. Video-conferencing

CU-SeeMe runs over TCP/IP, but it is basically the same as any video-conferencing system running through ISDN lines, leased phone lines, or Multi-point Control Units. [14] A video-conference is very much an enhanced telephone call. However, the computer affords a higher level of control over the way in which the communication proceeds and is perceived by each participant. Most video-conferencing systems like CU-SeeMe have not generally striven to offer newer or better methods of communicating. They usually increase the immersion of each participant into a dialogue where physical proximity is not possible, but until recently, have concentrated mainly on the *tele-* part of tele-communication.

> "...we argue that a better way to solve the tele-communication [problem] is not to focus on the *tele-* part but the *communication* part. ...If we ever hope to solve the telecommunications problem, we must develop tools that people prefer to use even when they have the option of interacting in physical proximity as they have heretofore. To do that requires tools that go beyond *being there*." [10]

Many video-conferencing systems such as Intel ProShare®[3] are now emerging offering a wider range of communication methods than the simple "video phone" model. Some allow individuals to share applications while others allow a more interactive type of conversation through the use of whiteboard drawing spaces. As video-conferencing technology becomes more commonplace, new and better ways of putting it to practical use will certainly emerge.

2.1 Making the Connection

The first step in making use of video-conferencing is to find other relevant parties with the same capability. Generally, you must contact the other party via e-mail or by phone to set up the video-conference. Individuals can leave the system running on their computer, which makes it possible to "drop-in" on the individual. However, unless they are in front of the screen and ready for your

---

[3] From an online review by John Martell <martell@ucs.ubc.ca> on the Cornell listserver CU-SEEME-L@cornell.edu

call, you need some method to get their attention. The convenient familiarity of a ringing telephone, answering machines, and returned calls is certainly ideal when it comes to video-conferencing but is not yet commonplace. Face-to-face, synchronous communication is not always appropriate to the situation, so what we need are more tools to improve the flexibility of making video-conference connections.

Individuals have their own ways to signal to others if they are ready for communication. They are subtle but well practiced and can control the engagement or avoidance of a conversation with neither party being fully aware of what has taken place. [12] One can easily avoid a phone call by not answering the phone, or letting the machine answer the call. This allows the receiver full control over accepting the connection without having to completely accept the responsibility for refusal; they might be "out" as far as the initiator is concerned. Establishing connections with most video-conferencing systems, however, is out of the hands of the receiver or must be explicitly accepted or rejected by the receiver. Simply popping up on someone's screen is found to be rather intrusive by most people, and there are multiple concerns about privacy when someone can just pop in and watch you. [8] So, making a video-conference connection is a deeper issue than simply connecting two computers. It is *people*, not computers that are connecting in a video-conference, and what we need is a variety of well-established communication-initiation methods that serve the caller while maintaining the privacy and control of the receiver. [6]

2.2 Conducting the Conference

Computer-based video-conferencing systems certainly open exciting possibilities and new channels for communication, but they also tend to introduce their own set of difficulties. With a video-conferencing system, regulating the conversation between participants becomes problematic due to the asymmetry introduced into the communication. [1] When talking on the telephone, for example, parties rely solely on language cues like pauses to regulate the

conversation and each party generally has the same perception of the current state of the conversation. In person, conversants -- even in groups -- will usually have predictable patterns and cues that regulate the conversation. [21] However, when we introduce video into the conversation, regulation of the conversation becomes more difficult. The physical cues are eliminated at the receivers end, but the speaker may still be relying on the cues. For example, when using CU-SeeMe the camera is generally placed at a far enough distance from the user's focus on the screen that when the user is looking at another party on the screen, they have the natural feeling that they are looking directly at the other person. However, the second party will see them looking away and will not have the same perception that the speaker does. It is this asymmetry that causes the problem and not the simple fact that the conversation is video mediated. [21] [22]

2.2.1 Video-conferencing Topologies

There are three common topologies that model the ways to conduct video-conferencing.

I. Point-to-Point

A simple point-to-point connection is much like a regular telephone call. One computer waits for the connection and another computer "calls" the waiting computer. In a system such as CU-SeeMe, nobody else can join-in on a point-to-point conference. This fact is due to the manner in which the system makes connections. Systems like CU-SeeMe allow only one connection from each computer which means that to connect to more than one party requires the use of a central hub. A system in which one can make multiple point-to-point connections without the use of a central connection point offers much more flexibility in terms of conferencing. Allowing multiple point-to-point connections from a single computer makes it possible for all the users to remain in an active *waiting* state at all times, e.g., even when they are connected to someone, they can still receive an incoming video-call from someone else. If they are connected to a central hub and

engaged in a multi-party conference, they could still receive a point-to-point connection from someone else.

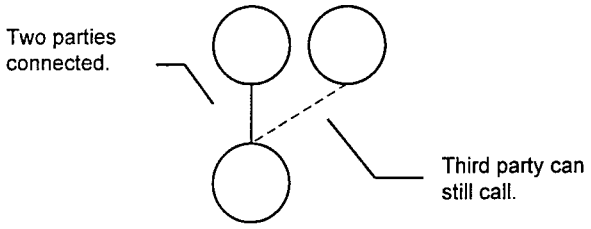


Figure 1: Point-to-Point multi-party conference

With a small number of participants (4 or less), this scheme does not significantly increase the number of signals each computer must process during a multi-party conference and it offers more flexibility in the conduct of video-conferencing. In Figure 1, the party receiving the request for a second connection would now have two parties on their screen, while the other two parties would not be connected to each other. Not automatically connecting the two second parties to each other reduces the amount of signaling over having all three connected point-to-point while allowing the first party to talk to two people at once. The first party would be able to selectively talk to either second party, or talk to both simultaneously. The ability for the first party to then connect the two other parties to each other would be helpful but may prove too cumbersome in practice for most packages. However, since each party could open multiple connections, they could all agree to open another connection to a common multi-connection hub, like a CU-SeeMe Reflector, and once established they could terminate the original point-to-point connections. This kind of "video call waiting" could be very useful in an educational setting where a teacher could give an oral test to two students simultaneously and maintain separation of the students.

II. Centralized Connection Point

Some systems use a central hub, called a Reflector in CU-SeeMe, to which all parties connect in order to conduct a multi-party conference. Two or more parties connect to the hub, each sending

a single signal to the hub. The central connection point then broadcasts all of the signals to all of the connected parties.

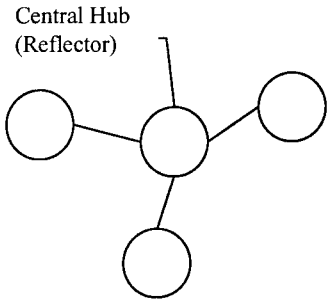


Figure 2: Multi-Party Connection with Central Hub

Once a multi-party conference is underway using a central hub, new parties can "pop" in anytime and communicate with all of the currently connected parties. Also, new parties can connect to the hub without sending a signal and "spy" on the conference. Each individual in the conference can then selectively turn on or off participants on their own screen. This scheme saves on computer and network resources for a multi-party conference since each computer only needs to send one signal. CU-SeeMe can only handle a single connection for each computer. Once connected to a Reflector, the only way someone else can "call" you is to connect to the same Reflector (and they probably will not know where you are connected). Again, this is where allowing the system to open multiple connections would be helpful. Even if the system is limited to only two connections, it is much more flexible from a communication standpoint. It is the same concept as "call-waiting" on a telephone but both connections are active simultaneously. Opening multiple connections is distinct from a multi-party conference. For example, CU-SeeMe allows a multi-party conference but only one connection.

III. Broadcasting

The NASA Select service is a good example of broadcasting through a video-conferencing system. This is simply a multi-party conference in which only one party, the broadcaster, is

sending a signal while all parties are receiving. We make a distinction here even though a broadcast is simply a multi-party conference in which only one party is *sending* a signal. Broadcasting is also enhanced by the ability of a system to open multiple connections. For example, an individual could be watching NASA TV without blocking out potential callers who may need to make a connection with that individual.

2.3 Communication Channels

There are various ways to communicate information with other parties in a video-conference. As stated earlier, this is one of the major challenges of the future of video-conferencing. These systems need to be much more than a way to establish a general presence. They must evolve into powerful *communication* tools allowing individuals to exploit a wide variety of communication channels beyond simple verbal exchange with video presence. Currently, there are a number of communication channels becoming available with video-conferencing systems and as more people use them we will discover needs for new channels.

2.3.1 Visual (Video)

The obvious communication channel in a video-conferencing system is video. Video frames come in various dimensions and resolutions. The frame rate is the speed at which the video picture is updated. Low frame rates make it difficult to use video for anything but a general presence. However, even a video-conference with a low frame rate can greatly enhance the communication experience as compared to a telephone conversation. Louis Lumière's slightly flickering cinématographe of 1895 ran at 16 frames per second, a 70mm film of today runs at 24 frames per second, [5] and CU-SeeMe connected to one other party will run 2-7 frames per second. The lack of eye-contact associated with the video picture in a conferencing system is distracting and can cause some irregularities in speech patterns. [11] [1] [22] However, having the video picture definitely adds an interesting dimension to any conversation.

2.3.2 Talking (Audio)

Audio capabilities mix naturally with video, but it is more difficult to make the quality of audio acceptable. Unlike video, when audio is slow or choppy it becomes unintelligible and useless. Compression schemes can greatly improve the quality of audio in a video-conference, but slow connections or lost data can quickly render the audio useless. Without smooth and acceptable audio quality, a video-conference becomes less desirable than a phone call.

2.3.3 Typing

The simplest way to communicate verbally, in lieu of audio, is to add a sort of "chat" capability between participants. Each participant has a text window or text line where they can type messages. Their text is then transmitted to the other participants along with their video. This makes it possible to "talk" over the video link without audio capability. Interactive "chatting" has been in place on the Internet for some time. One can establish a "chat" conversation in much the same way as a video-conference. Two individuals can establish a point-to-point connection or multiple individuals can all connect to a central location and conduct a group "chat", commonly known as an IRC group.

2.3.4 Whiteboarding

Whiteboarding consists of a blank drawing area that the parties involved in the conference all share. Each party can see the whiteboard on their own screen and can draw on the whiteboard. All the parties see the same board, so it is easier to communicate ideas that they cannot readily express orally. This sort of communication has been tried and tested in various forms but is probably easiest to implement with the help of a computer. [4] [11]

2.3.5 Application Sharing

This is similar to the idea of whiteboarding, but in this communication channel, the parties share a common application, such as a spread sheet, word processor, or presentation manager. Application Sharing is gaining momentum in the commercial marketplace with packages such as Intel ProShare®, AT+T Vistium Personal Video®, and IBM Person to Person®. [14] It has great potential for the future and is more flexible than whiteboarding in that, if you need a common drawing space, share a drawing program. Now, however, if you need to work on a spreadsheet together, you can also do that and still take advantage of the spreadsheet software's features. One person can take over the other person's computer to walk them through a task, teach a process, or simply illustrate an idea or concept that they find hard to express in writing.

2.3.6 Screen Transmission

This idea is a kind of "no-frills" computer-sharing concept. For a screen transmission, you outline a portion of your computer's screen that you would like to transmit as a video picture. Now, whatever you do in that space on your screen is transmitted to the other parties. The outlined portion of the screen is transformed to a normal video picture and transmitted. Issues such as varying screen sizes, and maximum sizes for the space to transmit should be relatively simple to resolve. CU-SeeMe has implemented a version of this concept in version 0.80b2 of their software. In it, you can display a static picture, or slide, in a window that is transmitted as a video window. You can then move your mouse pointer around inside the picture window which will be seen at the other end of the conference. (Unfortunately this update was released too late to make it into this Formative Evaluation)

3. Formative Evaluation

Although Formative Evaluation may not include rigorous, statistical testing of the software interface design, it is still quite formal and an extremely important aspect of software development. [9] Developers should perform Formative Evaluation early in the development cycle and continually perform it again as the interface evolves.

> "Users will evaluate your interface sooner or later....[so] why not do it right, and evaluate it sooner?"

CU-SeeMe is still under development, so this is the time to start seriously considering the quality of the interface, addressing any problems, and reinforcing strong points. This evaluation will include both critical and empirical reviews of the current interface. The critical review portion is a means to evaluate the system against HCI guidelines and uncover issues that may not be obvious through critical-incident analysis of test participants. Through benchmarks and critical incident analysis, The empirical evaluation provides insight into usability problems from the perspective of the target users. Each test participant will have the opportunity to comment both verbally and in writing through the use of structured interviews and quantifiable questionnaires. Both evaluations will be combined to produce a final list of possible ways to strengthen the CU-SeeMe interface.

3.1 Target Users

In the User Sophistication Taxonomy of M.L. Schneider, [19] the current batch of users for CU-SeeMe tends to span the entire spectrum. At the expert end of the spectrum are the CU-SeeMe developers and numerous individuals in the Internet community who have become part of the CU-SeeMe culture. At the other end of the spectrum are people like teachers and students who are more concerned with the usefulness of this software rather than its technical merits. Even the Parrot [19] user can be found glaring into the lens of a camera at video-conferencing demonstrations such as the one in the Exploratorium in San Francisco, California.[4] It is obvious

[4] As directly observed by the author on several occassions while using CU-SeeMe.

from the current interface that the focus of CU-SeeMe development, consciously or not, has been on the intermediate to expert user. The interface contains quite a few technically oriented features that may be irrelevant and confusing to the novice user who is probably concerned more with how easy the software is to use rather than its performance on the network. The novice will probably judge the performance of the software by the video quality and not packet losses or transmission rates.

It is the novice to intermediate user that I would like to concentrate on in this evaluation. Video-conferencing has the potential of becoming a commonplace communication medium for everyone, and not just the power Internet user. As an educational tool, the conferencing system must cater to the novice user so that it can serve individuals of all experience levels without interfering with the educational experience. It is important to understand that general attitude among many educators toward computers is that the computer is only a tool. There is always a percentage of the group who will desire to learn some of the technical aspects of Internet use. CU-SeeMe, however, needs to lend itself better to the novice user if teachers and students are to reap the benefits. Referring to the technical aspects of the Internet and most Internet software, Melissa Matusevich, a teacher with the Montgomery County school system and one of the local pioneers in bringing the Internet to teachers stated that, "Generally, most teachers do not know much about this."[5] The novice user is still only one part of the entire user audience of CU-SeeMe. Expert functionality should still be included to facilitate the fact that this area of technology is still under constant improvement and development -- much of which comes from suggestions or critiques of many expert users.

3.2 Critical Review

In this study an empirical evaluation is planned and will serve as the main source of data on the quality of the current CU-SeeMe interface. Subjective user opinion, critical incident analysis [2] [7]

[5] Matusevich, Melissa. E-mail message "Re: Resources Knowledge", Mon 12 Sep 1994.

[13], and benchmark tasks [3] [9] should uncover the most critical problems with the system, however, less obvious or rare incidents cannot always be covered in an empirical benchmark design. A critical review will allow us to view the system under the guidance of some established Human-Computer-Interaction guidelines. [23] Instead of focusing on critical incidents and benchmarks, we can essentially "pick apart" the interface using a subset of the Smith and Mosier HCI guidelines [23] to steer the process.[6] Since this review is performed by an expert user experienced with CU-SeeMe and is driven by a subset of HCI guidelines, we may identify issues that user performance in the empirical study may not uncover. This review will be used to augment the empirical results and is not intended to be an authoritative source of identifying usability problems. It will be looked at in retrospect after completing the empirical study. The goal is to see what problems were predicted that actually occurred with users in the empirical study and which predicted problems did not occur. Potential problems stated here which are not pinpointed through the empirical tests may be simple enough to fix that they should still be considered. It may be the case that the users did not have opportunity to experience every problem since in reality we are limited to how long we can keep each participant willfully participating. The empirical tests focus on the users' performance identifying specific problems actually encountered that should be fixed to make the software more usable. This review simply focuses on the quality of the interface alone by identifying items that *should* be fixed to make the overall quality of the program better. The goal of any product should not be to simply make it useable, but to make it something that users really enjoy using.

3.3 Critical Review Method

This review is a subjective review but by no means based solely on the views of one person. Most problems described here are those that have been pointed out by various users and then

[6] Reference Appendix A for a list of the guidelines used in this review.

simply reported.[7] This is indeed highly subjective since one individual may point out the problem they have and then another decides if it warrants being reported. The idea was to start with the observations of users and compare the source of the observation, or complaint, to the subset of guidelines in Appendix A. If the part of the interface that prompted the observation seemed to have violated the guidelines to any degree, it was pointed out in the tables. User complaints were looked at from any user willing to offer them, regardless of user demographic. Even the author participated by supplying some observations about the software from extensive personal use.

Table 1: Critical Review Table 1

Observation	Potential Problems	Solutions
There is currently no Users' Manual. There is a README file that does contain a system description, but it needs to be in more of a user-centered manual form.	Novice users will become frustrated if there is not a well constructed manual allowing them to go directly to the solution for any problem.	A well outlined and constructed manual that gives system information as well as instruction on "How-To" complete certain tasks. *-Help the user get started* *-accommodate user experience levels*
Many functions do not give any feedback when they are executed, like Stop Sending, for example.	The user may not be clear if they have initiated a command and may end up wandering themselves into trouble. This will lead to the user not being confident to explore the system.	Confirm operations that change the state of the system like Stop Sending, and offer the option to turn the confirmation on/off. *-prevent user errors* *-keep locus of control with user* *-make user actions easily reversible*

[7] Reported through the CU-SeeMe Listserver, CU-SEEME-L@cornell.edu and through various conversations with other CU-SeeMe users at various public Reflector sites.

Table 2: Critical Review Table 2

Observation	Potential Problems	Solutions
When connecting, if you uncheck *I will send video* and *I will receive video*, then no connection will be made, however if you connect, you can then disable sending and receiving video without disconnecting.	The problem here is that not all user desires can be predicted and, although it may seem unlikely, the user should be able to connect in any configuration they want. The user may be confused when they try to connect like this and with no indication, it simply does not work.	Simply allow the connection to be made even if the user does not send or receive. Or, don't make the connection as before, but provide feedback that no connection can be made unless sending or receiving. *-keep locus of control with the user* *-use informative feedback*
Connect... and Connect To> are redundant on the Conference menu.	This can lead to confusion with novice users as to which one to pick. Also, both options are the same action but have been unnecessarily broken apart.	Use only Connect To> but above the *Self* option, put a *New...* option. When the user selects *New* they can make a manual connection and then be given the option to add the site to their Nickname list. *-optimize user operations*
Text marquee is a useful function for displaying short messages, but can sometimes be hard to read. Also, you cannot display long messages on the marquee.	Users may need to display scrolling information to others in a conference or may wish to have a greeting displayed. If the text is unreadable, this function becomes useless.	Some users have taped a small strip of paper to the lens of their camera to give contrast to the text. This can be done in software by putting a black strip in the video window as text background. Put a marquee button on the local window that allows the user to dbl-click and construct a marquee message in a dialogue box and then start/stop it by clicking. *-organize the screen to manage complexity*
The info button should give more info.	It can be a waste of time exchanging information that can easily be available at the click of a button.	Increase the amount of information that can be displayed in the information box.

Table 3: Critical Review Table 3

Observation	Potential Problems	Solutions
Text marquee is difficult to use for chatting. Long messages are difficult to relay accurately and the text is often washed out by the background or broken up during slow video transmission. Also, when the video is mirrored the text is backwards. The marquee is a very unnatural way to converse using text.	When audio capability is non-existent or not working well due to network congestion, text is the only way to keep the conversation going. Without strong verbal communication capability, the videoconference becomes a bunch of people looking fish-eyed at each other. When the text is washed-out, or broken up while scrolling off the screen, it becomes useless.	The use of an IRC-style chat window instead of sending conversational text as video. Each participant could have their own two or three line text window under their video and then the IRC window could echo **selected** text streams to make it more conversational. This way the user is not bothered by unwanted text in the IRC window but can still see what others are typing. The IRC window is simpler to use when typing a multi-way conversation. *-keep it simple* *-optimize user operations* *-cognitive directness*
Technical aspects of the videoconference like software and hardware performance, network statistics, and compression/transmission parameters are too apparent. For example, the black border around a video window to indicate the Quickdraw routine is being used is completely irrelevant to most users.	Technical functions, displays, and parameters are an important part of the CU-SeeMe package but do not relate to the task of video conferencing. Novice users can become confused or overwhelmed by displays that they do not understand. Always displaying technical terms or statistics can scare away users.	There needs to be a way to turn on and off technical aspects of the CU-SeeMe display. The program should default to only displaying conference information and should include a Debug... settings box where the user can then individually turn on or off parts of the technical display like bandwidth usage, frame rate, packet loss information, etc. Now, when the user turns on Debug, the information they have checked will be displayed. *-keep it simple* *-give user appropriate status indicators* *-accommodate user experience levels*

Table 4: Critical Review Table 4

Observation	Potential Problems	Solutions
Depending on the hardware installed, the program can start up in a variety of flavors. Sometimes with a video window, sometimes without. Sometimes with the audio window, sometimes without. For example, if the user does not have the hardware capability to send video, the local video window simply does not appear and the user is left with just a menu bar.	This leads to confusion if everything is not setup correctly. The user may be simply missing an extension but this will not be obvious by, for example, simply not showing the local window. The user has no real indication that there even is a problem and may just wonder why that particular window didn't show up. They may then search to try and open that missing window to no avail.	Startup should always look the same in terms of windows open, etc... unless specifically changed by the user. Always open the local window and the audio window, but disable parts that are not available or put a message in the video window indicating video not available. Also, display a message on startup with a "Don't display this message in the future." button that tells the user what is missing if they are in a less-than-full configuration. *-Help the user get started* *-Accommodate individual user experiences and differences.* *-Use informative feedback* *-Use appropriate status indicators*
The button bars on both windows are not repeated in menus. There is a "button bars on" mode and a "button bars off" mode.	If the user has the button bar turned off, there is no way to access those functions except to go into the menu, turn on the button bars, then use the button.	All the functionality of the system should be accessible in the menus. Add the functions provided by the button bars to the menu system. *-optimize user operations* *-use modes cautiously*
Both windows contain an info line, but you can't remove the one from the second party window. Also, the info lines differ in their display. What they display must be more consistent.	If a user may want to get rid of the local info line, then they may also wish to get rid of the second party info line. Also, it can be confusing as to what is meant by some of the messages since they differ slightly between the local window and the second party windows.	Make it possible to get rid of the info line in the second party window. Standardize and minimize the messaging in the info lines as much as possible using identical terms for identical states. *-be consistent* *-keep it simple* *-use informative feedback* *-give user appropriate status indicators*

Table 5: Critical Review Table 5

Observation	Potential Problems	Solutions
The eye-speaker-mic metaphors don't match especially from an ownership standpoint. Also, the eye isn't even a button but it is located on the button bar and resembles a button.	Might be confusing to users as to who the icon refers to...the local user or the second party. The eye would seem to belong to the person in the window indicating they can see you, so does the microphone belong to them indicating they can talk to you? Mixing buttons and status indicators can be confusing.	Move the location of the eye status indicator. Put it in the status info line below the button bar. Moving the eye may reduce confusion over the speaker and microphone buttons since the three will not be mentally grouped by the user. *-draw on real world analogies* *-give the user appropriate status indicators*
Menu system is lacking access to many functions.	This leads to users searching aimlessly for a desired action. If a user does not see a function afforded to them right away, they will probably hunt through the menus and if its not accessible in the menus they may decide its probably not available.	Every action should be included in the menu system. For example, simply adding access to the settings via the menu bar would save many users from hunting and then the actions could be assigned shortcut keystrokes for the power user. *-optimize user operations* *-recognition rather than recall*
When picking a connection from the Connect To> list, a second window pops up to initiate the connection which seems unnecessary.	Users may prefer that the selection lead directly to the desired action.	Allow an option to turn on or off "confirm connections". Seldom will the information in a Nickname change and if a conference ID is needed, the system can then prompt for one, so when a name is picked from Connect To>, initiate the connection without confirmation, unless the preference is set to confirm. *-keep locus of control with the user* *-optimize user operations*

Table 6: Critical Review Table 6

Observation	Potential Problems	Solutions
Audio controls and status indicators are not very clear. The Push to Talk mode is confusing.	It can be very confusing trying to ascertain who is currently speaking. Scanning the open windows to find out who is talking doesn't work because by the time you find it, they're done talking. Also, it may be difficult for users to figure out how to use the Push-to-Talk mode since there is nothing to push.	There needs to be an obvious indication of who is speaking that can instantly draw the eye to the proper window without being too distracting. The audio window should be streamlined with a large button that the user can actually PUSH to talk. Instead of a "Push-to-Talk" mode button, allow the TALK button to be locked in the pressed position for continuous audio. *-optimize user operations* *-use informative feedback* *-give user appropriate status indicators*
Error messages are extremely non-descriptive.	Even expert users will have a problem here if there is a system error that reads something like, "Error -227". Most users will probably not even attempt to figure out what the error message means.	Make error messages that are descriptive. Even if an error flag indicates more than one possible error, indicate that fact and list the error possibilities. *-keep it simple* *-use informative feedback* *-use specific, constructive terms in error messages*

3.4 Empirical Usability Evaluation

To evaluate a piece of software in a user-centered [9] way means that we must first decide how a user would in fact use the software. Instead of constructing benchmark tasks to measure parts of the interface that we see as the most important, it seems more logical to model scenarios of how the user would use the software in practice and then extract from those scenarios the parts of the interface that we should measure. This way, not only is the testing user centered, but the design of the tests is user centered as well. Once we have a logical set of scenarios, we can extract specific tasks from those scenarios which can be used as benchmarks which can then, in turn be used to maintain a set of measurable usability specifications. [3] [9] [24] Usability specifications are useful in the iterative development process [9] since we can use them repeatedly and quantifiably measure the progress of the interface design. Scenarios should also make the collection of critical incident [2] [7] [13] observations more useful as well since the participants will be performing tasks in a manner that more closely matches actual usage. We will collect data on critical incidents in order to gain a more user centered viewpoint about the CU-SeeMe interface. Simply collecting data on *where* the problems occur is insufficient if we do not understand *why* the problems occur. Observing the users for possible critical incidents, and then allowing the users to confirm *where* they had problems, exactly *what* caused them and *why* is crucial to understanding *what* needs to be modified in the interface and *how* it should be modified. The quantifiable measures can pinpoint *where* the problems are and critical incidents can help identify exactly *what* the problems are and *why* the problems are indeed problems.

3.4.1 Empirical Testing Methods

I. Test Scenarios

To develop scenarios for CU-SeeMe required observing how some individuals are using the software and then using that information to derive a set of likely scenarios for usage in a

classroom. Knowledge of the capabilities of the software, and why it might be used by a particular set of users can be combined to form the scenarios. These scenarios are not observed scenarios but must be creatively and logically formed in order to model what we see as probable practical applications of the software. As the software becomes more widely used, actual stories can be collected, reviewed, and combined to form more scenarios. This set of scenarios was derived through conversations with local school teachers. The teachers were asked how they see themselves using a package such as CU-SeeMe, and how they might apply it in the classroom. From these talks and some creativity, came the following scenarios for CU-SeeMe usage in a classroom setting.

Scenario 1

Mrs. Applebee, a rural elementary school teacher, has just received a new computer in her classroom that has a network connection, audio and video capability, and a copy of CU-SeeMe installed. She does not yet have a camera or microphone, but she will be receiving both in the near future. One of her colleagues, Mr. Math, has an identical setup in his classroom and has already received his camera and microphone. Mr. Math's elementary school class is located in the city and Mrs. Applebee wishes to start an inter-class relationship with her class and Mr. Math's class. She hopes this will let the children in both classes better understand the other. Mrs. Applebee contacts Mr. Math and arranges to try out CU-SeeMe even though she currently has no video camera or microphone. She should be able to get at least a black window so that she can type messages.

At the appointed time, Mrs. Applebee launches CU-SeeMe and establishes a connection with Mr. Math. She types a greeting and her IP address to Mr. Math so that he will have it for future reference. Mr. Math says "Hello" and gives her the date and time of their next meeting. Mrs. Applebee cannot find the piece of paper with Mr. Math's IP address on it and wants to add Mr. Math to her Nickname list. Mr. Math tells her to click the information button on his window and

copy it from the information box. Mrs. Applebee then adds Mr. Math to her Nickname list so that next time she can select his name off the Connect To... list. They each say "Good-bye" and disconnect.

Scenario 2

This is the second meeting between Mrs. Applebee and Mr. Math. Mrs. Applebee now has a full setup and wishes to test it. Mrs. Applebee starts CU-SeeMe, adjusts her video camera, and then adjusts her video picture brightness and contrast to suite. The school administration has told her that she must conserve bandwidth and set her audio for the least bandwidth usage. She sets her audio to the Δ-mod compression scheme since someone told her that was the lowest bandwidth scheme. She connects to Mr. Math and must get his attention since he has his back turned to the machine. She explains to Mr. Math that the first thing she would like to do with the classes is to have each class share their "Show-and-Tell" with the other class. Mr. Math agrees that his class will go first and sets the appropriate date and time. They disconnect their conference and Mrs. Applebee cleans up her desktop.

Prior to the first "Show and Tell" meeting, Mrs. Applebee wishes to configure her CU-SeeMe package so she will not have to fuss with it the day of the video-conference. She makes a connection to herself, places the second party window in the center of the screen and enlarges the window since she will be projecting the screen on the overhead projector for her students to see. She checks all her options and adjusts the transmission settings to control her bandwidth use according to the directive from the school for teachers using CU-SeeMe in the classroom during school hours.

Scenario 3

With the excitement of the video-conferencing system growing, Mrs. Applebee decides to bring another party into a conference. She arranges for a speaker to give a talk to both her and Mr. Math's class. At the appointed time, all three connect to the Reflector, everyone introduces

themselves, and the speaker begins talking to the students. Mrs. Applebee establishes a private talk channel to Mr. Math so she will not interrupt the speaker and tells him to stop sending his video so the speaker's image will be more fluid. She disables the private channel but can hear background noise from Mr. Math's class since he has his audio in continuous transmit mode. She disables his audio so that her class can only hear the speaker. Mrs. Applebee closes the window for Mr. Math's class, sets her system to not send, and closes her local window to ease the distraction from the speaker and lighten the load on the network so the speaker's signal will come through better. During the talk, the speaker asks if there are any questions and someone from Mrs. Applebee's class asks a question that she repeats to the speaker. Before repeating the question to the speaker, she reopens her local video window and resumes sending video. She also reopens the window for Mr. Math's class. She ensures that both Mr. Math and the speaker can see her before speaking and then puts her audio in continuous mode so that the class can be heard. The speaker disconnects from the conference, Mr. Math and Mrs. Applebee say "Good-bye" and they both disconnect.

II. Benchmark Tasks and Usability Specification Tables

CU-SeeMe is still early in its development cycle and has not yet undergone Formative Evaluation. Because of this, metrics to quantify usability specifications do not exist on the actual product so we must "best guess" where to begin. CU-SeeMe contains a fairly shallow interface in that the functionality of the system is rather specialized and does not require many deep level commands or controls. Because of the relative simplicity of the interface, we will use qualitative data based objectively on critical incident analysis [2] [7] [9] [13] and subjectively on structured interviews as the primary means of evaluating the software. However, relying solely on interviews and observations would make it more difficult to measure progress in a re-evaluation. With some quantitative data, further testing with the same benchmarks can be done in the future and compared to the initial testing on the basis of these numbers. This will allow an objective view of

how the interface is improving. All values chosen for the specifications have been logically estimated based on experience using CU-SeeMe and inputs from pilot testing.

From the scenarios described earlier, we first extract a list of physical actions to perform with CU-SeeMe that follow the scenario descriptions as closely as possible. Next, these physical actions are separated into groups of actions that have a high degree of closure. There must be a definite, observable beginning and end to each measurable task. [3] [9] These separated tasks are used as the benchmark tasks, and then modified for clarity or time constraints during pilot testing. Each benchmark task is then associated with a set of measurements like time, error count, and user satisfaction and we can produce Usability Specification Tables. [3] [24] Choosing values for Current Level, Worst Acceptable, Planned Target, and Best Possible is difficult for these tests since testing has never been performed. There is no prior data evaluating the CU-SeeMe interface and therefor we must use logical "best-guesses." Each benchmark has listed with it a comparable action or process that was used to estimate *Current Level* for times and errors. These are only estimates used to give momentum to the specifications and can be updated for successive iterations of the Formative Evaluation. *Best Possible Level* was estimated using the actual times of two expert computer users who have extensive experience using the CU-SeeMe package. *Worst Acceptable* and *Planned Target* were then estimated at reasonable levels based on the *Current Level* and *Best Possible Level*. For time measures, the *Planned Target* Level started out by doubling the expert users' time and then adjusting it slightly to a reasonable level considering the comparable task. Error counts do not include simple typing errors or Macintosh-specific errors. Each Specification Table follows with the associated benchmark task listed above. The benchmark tasks are written out explicitly and will then be transformed into a set of user instructions which eliminate the *how* and only include the *what* to do.

Usability Specification 1

Reference Scenario: Scenario 1

Benchmark #1: Conference without video or audio
Start the CU-SeeMe program by double-clicking on the icon. Select Connect... from the Conference menu. Type in the IP address of Mr. Math's machine, ensure the send and receive boxes are both checked, and then click on Connect. When Mr. Math says "Hello", type a greeting and the address of your machine to Mr. Math and ensure that he received the address correctly. Ask Mr. Math when you should meet next and write this down on a piece of paper. Next, ask Mr. Math for his IP address and write it down. Next click on Edit Nickname in the Edit menu. Add Mr. Math to your Nickname list by entering his name in the Nickname field, and IP address in the IP address field. Fill both the check boxes "I will send video" and "I will receive video". Click on OK. Say "Good-bye" and disconnect the conference by choosing *Conference - Disconnect* from the menu. Close CU-SeeMe.

This task can be compared to an IRC "chat" conversation. Note this is the first exposure to the software.

Table 7: Usability Specification 1, First Connection (no video or audio) [24]

Usability Attribute	Measuring Instrument	Value to be Measured	Current Level	Worst Acceptable Level	Planned Target Level	Best Possible Level
Initial Performance	Benchmark #1	Time to complete.	3:00	6:00	3:00	2:00
Initial Performance	Benchmark #1	Number of errors or wrong selections.	N/A	3	1	0
First Impression	Questionnaire #1	Average score.	N/A	6	8.5	10

Usability Specification 2

Reference Scenario: Scenario 2

Benchmark #3: Setting video, picture, and audio
Start CU-SeeMe by clicking on the program icon. Adjust the camera so that you are centered in the local video window. Bring up the settings box on the local window and choose Picture from the dropdown list. Adjust the brightness and the contrast to acceptable levels.
Choose Connect To... from the Conference Menu and select Mr. Math from the Connect To... list. Click OK in the Connect Box. Once you have connected to Mr. Math, tell him you would like to set up a "Show-and-Tell on Wednesday" and ask him what time is good. When he gives you the time, repeat it back to confirm. Say "good-bye" and disconnect the conference. Close Mr. Maths window and close the information and settings boxes on your local window.

This benchmark task will be compared to a phone conversation in which the initiator sets the phone volume, selects speaker phone, and conducts the same conversation done in this task.

Table 8: Usability Specification 2, Second Conference (with video and audio) [24]

Usability Attribute	Measuring Instrument	Value to be Measured	Current Level	Worst Acceptable Level	Planned Target Level	Best Possible Level
Initial performance	Benchmark #2	Time to complete	3:00	6:00	3:00	1:15
Initial performance	Benchmark #2	Number of errors or wrong selections	N/A	3	1	0
First impression	Questionnaire #2	Average score	N/A	6	8.5	10

Usability Specification 3

Reference Scenario: Scenario 2

Benchmark #3: Setting compression, transmission, and audio
Choose Connect To.. from the Conference menu and select Self. Arrange the windows side by side and then choose File from the menu bar and select Save Window Positions. Open the settings box on the local video window. Ensure the screen is set to Standard Resolution by choosing the Compression settings box on the local window. Next, choose Transmission from the settings box and set the Min. Kbits/sec to 20, the Max. Kbits/sec to 90 and the Max. Frame Rate to 25. Choose the audio settings box and set the audio to Δ-mod (16kb/s). Close the settings box on the local window. Disconnect from Self and close the second window.

There is no realistic comparison to this benchmark.

Table 9: Usability Specification 3, Configuring and Repeat Performance [24]

Usability Attribute	Measuring Instrument	Value to be Measured	Current Level	Worst Acceptable Level	Planned Target Level	Best Possible Level
Initial performance	Benchmark #3	Time to complete	N/A	4:00	2:00	1:00
Initial performance	Benchmark #3	Number of errors or wrong selections	N/A	3	1	0
User satisfaction	Questionnaire #3	Average score	N/A	6	8.5	10

Usability Specification 4

Reference Scenario: Scenarios 3

Select the Conference menu and choose Connect To... Pick "Info Reflector" from the list.

Benchmark #4
Once all three parties are visible, arrange the windows so that they are in a row. Establish a private audio channel with Mr. Math by clicking on the microphone button on his window. Ensure that he can still hear you by asking him to respond. Ensure that the third party cannot hear you by typing "can you hear me, speaker?" also saying this aloud. Close the private channel by again clicking on the microphone button on Mr. Math's window. Now, disable Mr. Math's audio by clicking on the speaker icon on his window. Ask Mr. Math to say Hello and ensure that you cannot hear him. Ask the speaker if he heard Mr. Math. Enable Mr. Math's audio by again clicking on the speaker icon on his window. Close the local video window and choose Stop Sending from the Conference menu. Next, close Mr. Maths window by clicking in the box in the upper left corner of the window. Now, move the speaker's window to the middle of the screen and enlarge it by clicking in the zoom box in the upper right corner of the window.

The comparison to this benchmark is a conference-telephone call. This comparison, however, is not very accurate since it is lacking functionality compared to CU-SeeMe.

Table 10: Usability Specification 4, Multi-party with full video and audio, part 1 [24]

Usability Attribute	Measuring Instrument	Value to be Measured	Current Level	Worst Acceptable Level	Planned Target Level	Best Possible Level
Initial performance	Benchmark #4	Time to complete	4:00	8:00	4:00	2:00
Initial performance	Benchmark #4	Number of errors or wrong selections	N/A	4	1	0
User satisfaction	Questionnaire #4	Average score	N/A	6	8.5	10

Usability Specification 5

Reference Scenario: Scenario 3

Benchmark #5
Select Conference from the menu and choose Start Sending to resume sending your video. Ensure that the eye icon on the speaker's window is now open. Select Participants and choose Local video from the list to open the local window. Next, select Participants from the menu and be sure that Mr. Math does not have an X next to his name, then choose Mr. Math from the list to reopen his window. Ensure that Mr. Math can hear you by asking him to say the work "Yes" and then tell the speaker one of your students has a question. When the speaker acknowledges, ask the question "What kind of tea do you like at tea time?" After the speaker answers, say "Thank You, Good-bye" and disconnect from the conference. Make sure all the windows are closed except the local and audio windows and leave CU-SeeMe running.

This benchmark can be loosely compared to a multi-conference on a telephone just as in benchmark #4, however, the relationship between the two is not very precise.

Table 11: Usability Specification 5, Multi-party with full video and audio, part 2 [24]

Usability Attribute	Measuring Instrument	Value to be Measured	Current Level	Worst Acceptable Level	Planned Target Level	Best Possible Level
Initial performance	Benchmark #5	Time to complete entire benchmark	2:00	4:00	2:00	1:00
Initial performance	Benchmark #5	Number of errors or wrong selections	N/A	3	1	0
User satisfaction	Questionnaire #5	Average score	N/A	6	8.5	10

III. Identifying Critical Incidents

Critical incidents involve much more than a simple user error. They are events with a cognitive significance that a user encounters which greatly impede performance. [2] [7] [13] Simply pressing the wrong button, for example, is an error but may not necessarily be a critical incident. Continually pressing that same button because maybe its the only action that makes sense to the user may be a critical incident. Events that cause large performance deficiency or a breakdown of understanding are critical incidents to the performance of users. A group of users may erroneously press a given button on an interface expecting it to perform a certain function. If we view this error without identifying it with any type of critical incident, we may add the correct button to the interface without fixing the problem. The next set of users may still press this wrong button even though the correct one exists. By noting the error the first time around and then getting feedback from the users as to *why* they committed the error, we may find a breakdown of understanding and identify this as a critical incident. With this, the root cause of the errors can be remedied. (Maybe it was a misleading icon)

For testing CU-SeeMe, this is the way we will try to identify critical incidents. A critical incident is in the eye of the user, so their input must be considered. During testing, the experimenter will observe the actions of each participant and note specific tasks that seem to be causing the most trouble. At the completion of each task, the experimenter will try to identify the critical incidents with the help of the user through a structured interview.[8] The notes taken during observation will help direct the interview, since a participant may not always recall each problem, but the input of the participant is the crucial ingredient. The observer may see the participant having a certain problem, like not being able to perform the next task, but may find out later from the interview that the participant was simply pressing the wrong mouse button or simply not paying attention (assuming they will admit to that). This interview data will be useful for identifying the importance

[8] See Appendix B, section 8.3.6

of each problem and formulating possible solutions to each problem. The user can explicitly state how confusing or unobvious a certain task was to them and how it affected their completing the set of tasks. This helps to identify the relative importance of each problem which will be reflected in the results tables. Also, the user can express *why* they had the problem and how the interface may have been modified to prevent that problem. The interviewer will try to lead the interview to pinpoint the *why* for problems by asking the participant what made them do what they did and why was it not more obvious to perform the task correctly. This helps to formulate a course of action to fix the source of the problem.[9]

IV. Pilot Testing

We conducted pilot testing using an expert user experienced with CU-SeeMe and an intermediate user who has never seen CU-SeeMe and never used a Macintosh interface. We chose the first pilot participant to help ensure the completeness and logical flow of the proposed participant tasks. The second pilot participant was chosen to ensure a complete protocol for testing as well as indicate any areas that may need clarification, like participant instruction sheets, questionnaires, interview questions or actual tasks to be performed. After the first pilot participant went through the tests, we modified many of the benchmark tasks. We needed to script the benchmark tasks since they would involve one or two other parties in each conference. The first pilot tester helped to make the scripts of each conference so that they flowed more naturally and closer to the scenarios used to derive the benchmark tasks. Also, hardware and configuration details were worked out with the first pilot participant. The second pilot tester served to further test the configurations of the three computer systems that would be involved in actual testing. Also, since the second pilot tester had no experience with CU-SeeMe it was possible to determine if the instruction sets were clear and with enough explanation for participants to complete each

[9] See the Critical Incident and Interview Analysis Tables in section 4.4 to view the problems, their estimated importance, and a suggested course of action.

task. We shortened some of the tasks after the second pilot test so that the entire experiment would remain within the desired time frame.

V. Testing Participants

Test subjects were chosen to match the target users of this testing as closely as possible. In this testing we did not use elementary school students because of time constraints and the extra protocol involved in using participants under 18 years of age. Subsequent testing should employ a population of elementary students but to do so, the scenarios from which all the testing was derived should be modified to match this particular population. According to empirical studies done by Nielson & Molich, the optimal number of participants for a cycle of formative testing is three to five per user class. We used five subjects in the empirical testing sessions, each accompanied by two other conference parties located at different machines in different rooms. Four of the five participants represented our target class of the novice to intermediate user. The fifth participant was included to see if there is indication of a difference in the types of problems users of different experience levels encounter. The two extra conference parties were not testing participants, but were experienced CU-SeeMe users. The following is a short profile of each of the five testing subjects.

- **Participant #1** is a 5th grade school teacher who considers herself slightly above the novice level of computer use. She uses computers daily in the school and visits the Internet approximately once a week. Internet resources that she uses include web browsers and electronic mail. She has minimal experience with the Macintosh platform and has seen CU-SeeMe one time at a demonstration.
- **Participant #2** is a school teacher who considers herself a novice computer user. She uses a computer approximately once a week and does not use the Internet. Her experience is primarily on the Macintosh platform and she has never seen nor used CU-SeeMe.

- **Participant #3** is a graphic designer who considers herself an intermediate computer user. She uses computers daily in work and uses e-mail on the Internet daily. She has little experience with the Macintosh platform and has never seen nor used CU-SeeMe.
- **Participant #4** is a teacher and a college student who considers herself a novice user. She uses computers and the Internet approximately once a week, primarily for web-browsing and e-mail. She has little Macintosh experience and has never seen nor used CU-SeeMe.
- **Participant #5** is a college professor and an expert computer user. He uses computers and the Internet daily, making use of all Internet resources. He uses the Macintosh platform infrequently and has used CU-SeeMe on the Windows/DOS platform.

VI. Conduct of Testing Sessions

Testing sessions were conducted in the Human Computer Interaction Laboratory at Virginia Polytechnic and State University. The tests consisted of three computers, a video camera, a video monitor, three CCD digital cameras, two conference "extras" and the testing participant.

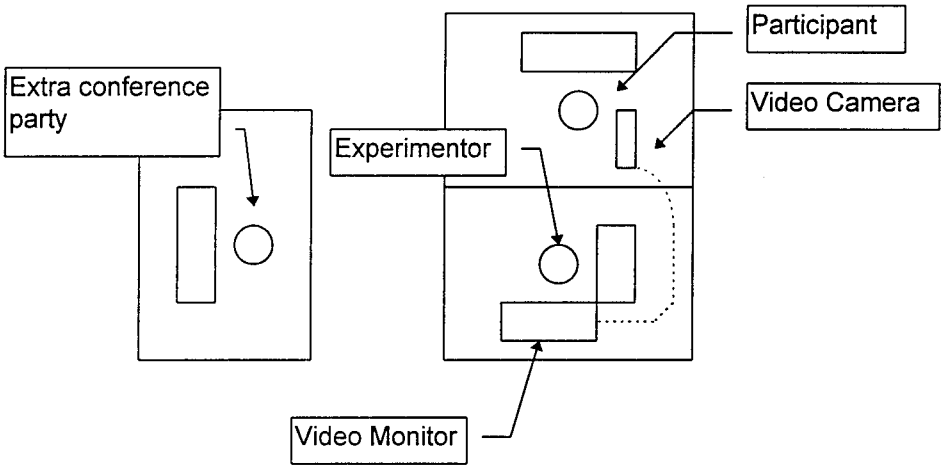


Figure 3: Testing Session Setup

Each participant was first explained the nature and purpose of the tests and given an Informed Consent Form[10] to read and sign. They were then given the User Instructions[11] and an

[10] See Appendix B, section 8.3.9

opportunity to have any questions answered. When the participant was ready, each of the five benchmark tasks were administered. The participant was given an instruction sheet for the given task and asked to read it entirely before beginning. When they were ready, a watch was started, the video tape started, and the experimenter left the room. The screen of the participant was monitored by the experimenter. When a participant started to make significant deviations from the task, the experimenter would enter and try to keep them within the scope of the tests without actually helping them complete the task. As little prompting as possible was given when needed for the sole purpose of keeping the experiment moving forward. Upon completion of each task, the participant would be asked to score a questionnaire and then answer questions in a structured interview. The purpose of the interview was to identify critical incidents that occurred during the task. The experimenter can observe the participants screen and attempt to identify critical incidents, but the only way to really know is to speak with the participant and discuss specific observations that may have been critical incidents.[12] After the interview, the participant would be given the next set of instructions and the next task would begin.

[11] See Appendix B, section 8.2
[12] See Empirical Testing Methods, section 3.4.1 part III

4. Empirical Results

4.1 Benchmark Results

Table 12: Benchmark Time Results

Part. #	Task #				
	1	2	3	4	5
1	18:13	6:30	3:47	5:48	5:29
2	10:51	12:29		10:12	
3	11:02	6:09	6:38	7:33	5:48
4	5:57	9:10	3:34	10:29	3:30
5	6:01	8:15	2:40	5:05	2:43
Average:	10:25	8:31	4:10	7:49	4:23

To get a real feel for where the problems lie, we must be able to see a complete breakdown of times measured. Since the testing population is small, we can easily identify outliers in the collected measurements and weight their value appropriately. Participants completed few of the benchmark tasks within the Worst Acceptable times specified in the Usability Specification Tables. Task 1 proved to be quite a challenge for less experienced users, except for participant 4 who claimed to be a novice user but still managed to complete Task 1 within the Worst Acceptable Time. It would seem that Participant 4 experienced a bit of "luck" since she did not consistently duplicate the performance achieved on Task 1. During interviews, some participants did express "luck" as an aid in completing the tasks. All the participants were fairly steady in performance except for Participant 1 on Task 1 (unusually high) and Participant 4 on Task 4 (higher than would be expected). On Task 1, Participant 1 paused several times trying to make sense of what to do next and spent some considerable time pressing the wrong mouse button. These higher time measurements usually correspond with higher error rates. (See Table 12) We considered time spent in extended pause as a critical incident[13] and combined all of the critical incidents according to common causes into the Usability Problem Tables.

[13] See Empirical Testing Methods, seciton 3.4.1 part III

Table 13: Benchmark Error Counts

Part. #	Task #				
	1	2	3	4	5
1	13	5	4	4	3
2	4	6	-	5	-
3	7	5	4	2	2
4	0	5	2	5	1
5	4	9	3	3	0
Average:	5.60	6.00	3.25	3.80	1.50

Again, with a small population anomalies in the data are easy enough to spot with a complete breakdown. The error counts in Task 1 were largely due to confusion over starting a conference and adding a Nickname. Task 2 produced fewer errors in the conference start, but now participants committed errors while they tried to use the audio for the first time. Poor visual cues left participants clicking the wrong button or not clicking at all to talk. Participant 4, who had no problem with Task 1, committed a number of errors trying to figure out how to use the audio. Participant 5 also had considerable trouble with the audio, clicking on the "Push to Talk" check box button several times to talk. Task 4 is the longest task but the error average is much lower than earlier Tasks 1 and 2. These three tasks are similar in operations to be completed by the participants and the fact that Task 4 is longer and still yielded fewer errors shows that CU-SeeMe does get easier with exposure. Once a participant figured out how to do an operation correctly, they generally did not repeat their mistakes. Both Tasks 5 and 4 resulted in average error counts that fall within the desired levels in the Usability Specification Tables. This is an indication of the learnability of CU-SeeMe due to the overall simplicity of the system but the error counts over all five Tasks indicates that the interface needs clarification.

4.2 Questionnaire Results

Refer to Appendix B for a complete questionnaire breakdown.

Table 14: Questionnaire Results

Participant		Task # 1	2	3	4	5
1	Mean	3.80	4.73	4.10	5.64	4.42
	Std. Dev	1.90	0.88	0.99	0.67	0.51
	Median	3.00	5.00	4.00	6.00	4.00
2	Mean	4.64	2.27	-	-	-
	Std. Dev	2.87	3.41	-	-	-
	Median	5.00	1.00	-	-	-
3	Mean	7.27	6.73	6.73	7.08	7.08
	Std. Dev	1.75	1.49	1.42	1.31	1.31
	Median	8.00	7.00	7.00	8.00	7.50
4	Mean	9.79	7.86	9.70	9.00	10.00
	Std. Dev	0.43	1.41	0.48	0.43	0.00
	Median	10.00	8.00	10.00	9.00	10.00
5	Mean	5.00	4.14	6.80	6.73	7.25
	Std. Dev	2.77	3.21	2.44	2.20	2.49
	Median	4.50	3.50	7.50	7.00	8.00
Overall	Mean	6.08	5.12	6.83	7.15	7.19
	Std. Dev	3.00	3.00	2.46	1.78	2.43
	Median	7.00	5.00	7.00	8.00	8.00

Task 2 did not fall within the Worst Acceptable Level prescribed in the Usability Specification Tables. All the other questionnaire scores, however, were within the Worst Acceptable although not at the Planned Target Level. It is important to note that at each successive task the scores tended to be higher with smaller deviations. As Participants became more comfortable with parts of the system, they seemed to like it better. Frustration became less as they learned some of the unobvious features and the general "feel" of the system.

4.3 Completed Usability Specification Tables

Table 15: Completed Usability Specifications, part 1

Usability Attribute	Measuring Instrument	Value to be Measured	Current Level	Worst Acceptable Level	Planned Target Level	Best Possible Level	Observed[14] Results
Initial Performance	Benchmark #1	Time to complete.	3:00	6:00	3:00	2:00	Mean: 10:25
Initial Performance	Benchmark #1	Number of errors or wrong selections.	N/A	3	1	0	Mean: 5.60
First Impression	Questionnaire #1	Average score.	N/A	6	8.5	10	Mean: 6.08 Std. Dev: 3.00 Median: 7.00
Initial performance	Benchmark #2	Time to complete	3:00	6:00	3:00	1:15	Mean: 8:31
Initial performance	Benchmark #2	Number of errors or wrong selections	N/A	3	1	0	Mean: 6.00
First impression	Questionnaire #2	Average score	N/A	6	8.5	10	Mean: 5.12 Std. Dev: 3.00 Median: 5.00
Initial performance	Benchmark #3	Time to complete	N/A	4:00	2:00	1:00	Mean: 4:10
Initial performance	Benchmark #3	Number of errors or wrong selections	N/A	3	1	0	Mean: 3.25

[14] See Tables 12, 13, 14

Table 16: Completed Usability Specifications, part 2

Usability Attribute	Measuring Instrument	Value to be Measured	Current Level	Worst Acceptable Level	Planned Target Level	Best Possible Level	Observed Results
User satisfaction	Questionnaire #3	Average score	N/A	6	8.5	10	Mean: 6.83 Std. Dev: 2.46 Median: 7.00
Initial performance	Benchmark #4	Time to complete	4:00	8:00	4:00	2:00	Mean: 7:49
Initial performance	Benchmark #4	Number of errors or wrong selections	N/A	4	1	0	Mean: 3.8
User satisfaction	Questionnaire #4	Average score	N/A	6	8.5	10	Mean: 7.15 Std. Dev: 1.78 Median: 8.00
Initial performance	Benchmark #5	Time to complete entire benchmark	2:00	4:00	2:00	1:00	Mean: 4:23
Initial performance	Benchmark #5	Number of errors or wrong selections	N/A	3	1	0	Mean: 1.5
User satisfaction	Questionnaire #5	Average score	N/A	6	8.5	10	Mean: 7.19 Std. Dev: 2.43 Median: 8.00

4.4 Critical Incident Analysis and Interview Feedback

A full review of critical incidents during testing, from video tapes, and from notes taken during test sessions uncovered a number of usability problems in CU-SeeMe. Many critical incidents were experienced by more than one user or were slight variations of identical incidents or causes. We have combined these duplications and variations to point out one common problem that spawned them. Problems identified both in the critical review and empirically should be considered seriously and even if the suggested solutions are not used, developers should create a solution following the guidelines included in the critical review tables. Possible solutions to each problem may not be entirely compatible with other problem solutions if they alter the function or structure of the system. Cosmetic changes like button re-labeling, status messages, or menu organization are generally low cost and should certainly be considered seriously. Issues of cost and resolution are ultimately left to developers to decide since only the development team truly knows the underlying system structure.

Table 17: Critical Incident and Interview Analysis, part 1[15]

Problem	Importance	Solution(s)
All users had trouble figuring out how to start a conference. Two users clicked in the local window and typed the address, expecting it to start a connection.	High	Supply a better visual cue to start a connection, like a button on the local window and supply better system state messages like "Not Connected-Waiting" to clue the user in that an action is necessary. Could relabel *Connect To>* to *Start Conference with>*.
Two users thought that closing the second party window disconnected the conference and waited thinking they were disconnected.	High	Supply a more obvious Disconnect cue like changing the Connect button (if added) to a Disconnect button and give more obvious status indication like "Connected to <IP address> or <Nickname>".

[15] See Empirical Testing Methods section 3.4.1 part III for a description of how this table was produced.

Table 18: Critical Incident and Interview Analysis, part 2

Problem	Importance	Solution(s)
One user had marquee scrolling with a space typed and then could not figure out why they could not type messages. Also, most users had trouble seeing the marquee text.	Medium	Make the marquee on a solid-contrasting background with a Stop, Single Arrow, and Double Arrow button to control scrolling.
Four users had trouble figuring out how to add a new Nickname to the list. (But most did not have trouble deleting since they already knew how to Edit Nicknames)	Medium	When a user connects to a new site, i.e. one not on the Nickname list, make the dialog box so that they can add that site if they type a name in the Nickname box above where the IP address goes. Also, relabel Edit Nicknames to Add/Edit Nicknames.
Two users found the information button on the second party window not totally clear.	Low	Get rid of the statistics button, (move access to that function into the technical area) and redraw the info button so it is a larger question mark or an "i" like displayed in some informational dialog boxes.
One user wanted Nickname added automatically upon request when connected to that location.	Low	Earlier solution will help here as well. Add the ability to add a site to the Nickname list when entering a new site to connect with.
Three users wanted Parrot-style "How To" section in a manual for common tasks.	High	Make "How To" section with major functions like Connect, Disconnect, Adjusting Video, Adjusting Audio, Typing Messages, Configuring Hardware, Talking to Others, Setting a Private Talk Channel, Disabling Someone's Audio, Opening/Closing Participant Windows, Opening/ Closing the Local Window, etc...
All users repeatedly clicked around in Audio window looking for settings and volume control.	Medium	Make the Audio window for control of actually speaking, i.e. Talk Button, Squelch Slider, and relabel the window "Speak" or "Talk" since it only pertains to that specific task.
Four users had trouble finding picture settings and even after finding them, had trouble finding the other settings layers.	High	Make all settings accessible from the menu except for Brightness and Contrast and redraw the button to more clearly convey the notion of Brightness and Contrast. (like a sun or the half-half circle)

Table 19: Critical Incident and Interview Analysis, part 3

Problem	Importance	Solution(s)
Three users had trouble figuring out how to re-open participant and local windows.	Low	Change *Participants* on menu to *Window*. Simply grey-out participants that cannot be opened.
Most users felt hesitant to explore, fearing they will get into an unknown state.	Medium	Confirmation of operations that change the system state substantially like Disconnect, for example will give the user confidence to try options. A Preference could be afforded to disable confirmations.
Two users confused a little between Disconnect and Stop Sending.	Medium	Separate the two functions. Disconnect refers to the Conference and Stop Sending refers to Video. Make an Audio/Video menu option that allows the user to control those items specifically. Also, better status indication like, Connected: Not Sending vs. Disconnected.
Four users went to Preferences... to change system settings.	Medium	Add an *Options* choice on the menu that contains *Configure...*, *Preferences...*, and *Debugging...*. This will allow the user to completely control their view of the system and the system can default to simplest view.
Three users searched in menus for access to some functions that were not included in the menus.	High	All functions should be included in the menus. Even items associated with a second party can be put in the menu and enabled when a second party window is the active window.
One user expected an X over the speaker icon when the audio was disabled, just like on the microphone icon.	Low	Make the display of icons consistent when disabled or unavailable, i.e. put an X over the speaker button when the party is not sending audio or the user has disabled their audio. Alternatively, gray all the buttons when disabled.
One user was not sure if the private channel had been established and did not know if someone was talking private to them.	Low	Better system status indicators. Like put a large P over the microphone to indicate Private and repeat the message Private Audio in a status line under the second party.
All users tried clicking on the "Push-to-Talk" check box to talk.	High	There should be a large button for the user to actually push when they want to talk and a check box to lock that button down (so it's always pushed). Only enable the squelch when the button is locked down.

Table 20: Critical Incident and Interview Analysis, part 4

Problem	Importance	Solution(s)
Three users tried using the Send and Receive buttons in the Audio window to control sending and receiving video.	Low	Get rid of these buttons in the Audio Window and put them in the Audio Menu.
All users could only figure out the private channel by chance from clicking on buttons and noticing the X's appearing on other windows' microphones.	Low	Balloon help on buttons that pops up when the user hesitates over a button would clear up confusion. Also separating status indication from action buttons would make the display less confusing. Like put a row of status indicators at the top of the window and make the button bar only buttons (that can be disabled when needed).
Some users often confused between Connect... and Connect To> and often made the wrong choice.	High	Get rid of Connect... and above *Self* on Connect To>, add the option *New*.

4.5 Overview of Some Interface Issues

Although the Usability Specifications stated in this evaluation were not set forth by CU-SeeMe developers, all the Target Levels were thought out carefully.[16] Some of the benchmark results measured during testing were very close to the planned target levels in the specifications. This is encouraging for a product that is still in Beta Testing, however, none of the planned target levels were met, which is an indication of the need for improvement in the interface. As CU-SeeMe gains in popularity, users will become more critical and will compare it with the many commercial packages currently arriving on the market. Most of the usability problems found through this evaluation do not require any remodeling of the system to fix, which should make implementing solutions feasible.

We cannot ignore the numerous merits of CU-SeeMe. The Cornell developers have brought a simple, yet powerful tool to the average Internet user. Many of the problems revealed in testing seem to stem from the fact that this is a highly developmental system. There are underlying technical issues to be resolved such as bandwidth consumption that far outweigh most of the interface issues. However, if the interface can be slowly "tweaked" along with these technical advances, the overall system will evolve smoothly.

The general attitude among the participants during testing was one of delight in the potential of CU-SeeMe. Each user felt that the interface needed enhancements, but the limited function of the system made it possible to overcome problems caused by the interface design. As CU-SeeMe evolves and developers add more functionality, users' ability to "overcome" the interface will disappear. The Analytical and Empirical Evaluations concentrated on searching out and locating problems associated with the interface design. The point of these critically oriented evaluations is to identify specific items in the interface that may cause problems with users and suggest specific

[16] See section 3.4.1, part II

courses of action for rectifying these problems. Now, using previously gathered data, we look at CU-SeeMe as a whole using the previously gathered data to support our evaluation.

4.5.1 Helping the User Get Started

Two of the testing participants completed Task 1 within a couple seconds of the Worst Acceptable Level set in the Usability Specifications. The remaining three users' measurements were not even close to the specifications. This trend continued in Task 2 with none of the participants meeting the worst acceptable time. Specific problems are pointed out in the analysis tables, but there is one idea that ties them together. CU-SeeMe seems to lack obvious visual cues to help users get started. In Task 1 most participants spent the greatest amount of time figuring out how to start a conference. In Task 2 connecting was not as much of a problem since the participants had already seen it once. Task 2, however, once again left the participants needing a better visual cue to get started, this time with the audio. Users had trouble getting the audio to transmit properly because they were either not clicking in the audio window or they were clicking on the "Push-to-Talk" check box to talk. Once they either figured it out or were prompted, participants had no trouble using the audio. It was simply in starting out where they needed more cues. Participants did not experience startup configurations that were missing elements like the audio or video window. However, if the computer system does not have all the required hardware and extensions installed for CU-SeeMe to work properly, it is not always clear why the program is not working properly. If the user does not know what the system looks like or how it works when installed properly, they may not realize that it is missing a component and will wonder why it is not working. There needs to be a complete set of startup messages that explain what is missing when the system cannot fully function. Video-conferencing with CU-SeeMe is as conceptually simple as making a phone call and should be so in practice, even to someone who has never done it before.

4.5.2 User Experience Levels and Learnability

There is no evidence in this testing to indicate that CU-SeeMe is easier to use for one type of computer user over another. Subjectively, the higher level users were more optimistic about the system but this is not reflected in the questionnaire results. Completion times tended to be lower for the one expert user but this is not a provable trend and there were not enough differences to warrant any kind of inference. Even error counts did not point to any differences between novice, intermediate, and expert users. Users from all classes seemed to have equal trouble getting started with CU-SeeMe and all were equal in how well they learned the features. In Task 2, the one expert user actually had the highest error count and was beat time-wise by a novice user and an intermediate user. CU-SeeMe is definitely a learnable system but this is due primarily to the fact that there is not very much to learn. Once a participant figured out an operation correctly they usually had no trouble with it on repeat performance of that operation. This was equally true for all participants and can be observed by reviewing tapes of each testing session.

4.5.3 System Status Indicators, Feedback, and the Mental Model

More than one problem arose during testing due to the lack of good system status messages and poor feedback. This problem was, however, offset by the fact that the mental model afforded to the user closely matches the system model. The purpose of the system is to send and receive communication signals, which are directly depicted in the local and second party windows. The actions that the user must take to complete a given task, like open a connection to start a conference, logically flow from a strong system model used by the developers. One participant noted, "Once you understand the model, you rely less on feedback, so the absence of it does not bother you as much." Although the mental model does offset some of the lack of status indication and stronger feedback, these two problems need to be resolved.

There are status bars on the local and second party windows, however, these do not display sufficient information for the user. Many menu choices, such as Save Window Positions, do not

give any indication that the operation was carried out successfully. Users were not sure if they were still connected when there was only one window open on the screen. Most participants wanted some indication of with whom they were connected. CU-SeeMe provides transmission statistics for the conference, but these are irrelevant to most users. When an expert user or developer is debugging the system or measuring performance, this kind of status information is important. However, for the novice to intermediate user who is concerned only with the use and not performance of the system, this information should be replaced by more relevant status feedback. When an operation is carried out that changes the state of the system, there needs to be an indication that it was successful. For some operations, like disconnect, which dramatically alter the system state, there needs to be confirmation feedback. These different kinds of feedback could simply be turned on or off in the *Preferences...* to accommodate different user classes from novice to expert and developer. Finally, the dual functionality of buttons on the second party window as both status indicators and buttons was confusing for some participants. One of the buttons is not even a functional button (the eye). As an example, when the user disables someone's audio, the button indicates that the audio is disabled by removing the little sound waves but shows no strong indication that it is the button to re-enable the audio. The status information should be grouped together and separated from the buttons. The status can still be reflected in some of the buttons by disabling or enabling them but now the focus of each button design can be the button's *function* and *state* (enabled or disabled) instead of some other system status information.

4.5.4 Keeping it Simple and Human Memory Limitations

The simplicity of the underlying system in CU-SeeMe makes up for some of the initial confusion with users by making it fairly easy to remember how to use the program once learned. Some participants needed experimenter intervention to complete certain tasks, but once the confusion was cleared, they could usually remember how to do the operation correctly during the next task.

CU-SeeMe does not contain operations that require users to sidetrack or navigate several levels deep and then return to the original operation. Operations are conceptually simple and straightforward and it was other factors such as lack of affordance and feedback that caused the most usability problems. The more technically oriented display elements like bandwidth usage, or transmission statistics complicate the program because they make the novice and some intermediate users feel uneasy. These display items are not needed for normal usage of CU-SeeMe. Although they should be available to accommodate higher level users, they should be "pushed into" the interface to shield the novice or intermediate user. The best way to insure that CU-SeeMe always remains a conceptually simple program is to default the system to its most basic function of conducting a video-conference and then let users discover more advanced features as they become comfortable.

4.5.5 Window Management

Window manipulation and window management become a problem in CU-SeeMe during large multi-party conferences. There must be better controls for managing windows that deal with opening, closing, and arranging. The *Participants* choice in the menu bar may be better labeled as *Window*. *Window* is a bit more direct in that you reopen a window that contains a conference participant; you do not reopen a participant. Users in testing did not have much trouble with management besides confusion over how to reopen closed participant or local video windows. During a multi-party conference with a large number of participants, windows can become cluttered. Unwanted windows will re-appear after the user closes them, and it becomes cumbersome to rearrange windows every time someone leaves or joins the conference. Preset window arrangements that the user can access via buttons or menus would solve this problem. If a desktop becomes cluttered, the user can simply choose an arrangement to instantly rearrange all the open video windows.

4.5.6 Navigation, Affordance, and Optimizing User Operations

CU-SeeMe is not a complicated system, which makes navigation simple. Some participants did, however, have considerable trouble finding certain parts of the system. These navigation problems were usually caused by lack of affordance. For example, functions accessible through the button bars are not present anywhere in the menu structure. A few participants would search through menus before randomly clicking on buttons, and when they did not find what they were looking for in the menus they did not always go directly to the buttons. Every participant had trouble navigating to the system settings. To find the Resolution settings, for example, the user must first go to the settings button on the local window, click on the settings list, select Compression, and then choose the desired resolution. The participants had some measure of trouble with this task because they could not find *where* to set the resolution. This navigation problem can be easily solved with a *Settings...* option in the menu structure or better visual cues in the settings layers (similar to the tabbed dialogue boxes used in some MS Windows applications). By simply increasing the affordance of many tasks, CU-SeeMe navigation will be greatly simplified.

Increased affordance of common tasks in CU-SeeMe will also serve to better optimize user operations. For example, adding a button to the local window to open or close a connection makes the most basic function directly accessible. Most CU-SeeMe tasks are already simple to carry out, but some can benefit from better optimization. When users pick a name off the Nickname list to open a connection, some did not understand why the second *Connection* window would pop up. Using Nicknames is supposed to eliminate the need to go through any kind of dialogue box since all relevant information should be associated with the Nickname. The idea of optimizing user operations should also be kept in mind when improving other aspects of the system like better feedback. For example, if modal message boxes are always used to give feedback instead of sound cues, optimization is moving the wrong direction. Shortcut keys are

already used for common menu items in CU-SeeMe and if the menu structure is expanded, this should continue since it optimizes the system for expert users.

4.5.7 Locus of Control

User feedback and confirmation are just a couple ways to increase the locus of control in the users' favor. From interviews it was learned that some participants felt threatened not by complexity, but by the feeling of not being in control. They were hesitant to select items because they could not "Take it back". This greatly hurts the interface quality because the user may feel a bit "bullied" by the system. One participant in particular, who represents a large portion of potential users, actually walked out of the experiment early, feeling belittled by the system. She made some menu choices, unsure if they were correct, and ended up putting the system into a state unknown to her. She did not know exactly what was going on and it was simply that she never had the option to back out, and had no strong indications that she was on the wrong track. It was the feeling that she was not in control that led to her extreme frustration. She stated that she would never use this in her class which is unfortunate because of the potential of CU-SeeMe as a teaching tool. For this type of user, the computer is a mere tool, to be controlled completely, and for the interface to work otherwise is disastrous. To put the locus of control in the users' hand, they must be allowed to "click around" without getting into trouble, and make the system work the way they prefer. Providing better feedback, confirmation, affordance of common tasks, and the ability to control a large number of user preferences is absolutely essential.

4.5.8 Real World Analogies

The term "Reflector" is a good example of where more real world analogies could help CU-SeeMe. This is a simple problem to fix - simply find a less technical name - but can greatly enhance the system in the eyes of the novice. The term Reflector is technical in nature, referring to the underlying aspects of the function of a reflector. Drawing on the real world, "Reflector" could be renamed something like "Conference Room" or even let individual reflector sites let their

name indicate their function, like "NASA Select TV", or their audience like "Mike's Coffee House". When connected to a given site, part of the system status indicator could show the max number of people that can be at that site, for example, "Maximum Seating 40. Seats available, 20". If the site is a point to point connection, just change the message to "Maximum Seating, 2. Seats Available, 0."

CU-SeeMe already makes some good use of real world analogies on the button bar in a second party window. Users liked the open/closed eye to indicate if the second party could see them or not but had a little trouble with the microphone and speaker. The eye leads to the notion that all the buttons refer directly to the individual in the second party window, i.e. their capability. However, this is not true and caused some confusion. Most participants figured out that the microphone indicated they could speak to the person in that window, but it was not obvious and some needed the manual or experimenter intervention. This problem, however may have been due more to the mixed analogies rather than the analogies themselves.

4.5.9 Error Messages and Error Handling ("Bulletproofing")

The testing did not uncover what is probably the most important feedback issue with CU-SeeMe. There is little error reporting on startup. CU-SeeMe must have certain items in the Macintosh Control Panels set correctly in order to work properly. If the panels are not set properly and CU-SeeMe cannot fully function, error messages will be displayed, but give no indication as to the possible cause or remedy. This is extremely frustrating for the novice to intermediate user who may be setting CU-SeeMe up themself and not have immediate access to any kind of knowledgeable support. Once running, there is little need for error messages in CU-SeeMe since there are not many syntactical type errors possible. Errors are usually in the form of erroneous choices that are valid, just not what the user intended. Bulletproofing CU-SeeMe is a matter of increasing the feedback to users in the form of confirmation dialogues. This can keep the novice

out of trouble and alleviate frustrations. Experienced users can use a simple on/off setting in the *Preferences...* to toggle confirmations.

5. Conclusions

We seemed to have uncovered some major usability problems with CU-SeeMe. Difficulties performing the most basic tasks like starting a conference and using the audio occurred with every testing participant. Although the numerical data suggests that the participants became more comfortable with the system as they progressed along in the testing, they were still having troubles at the end. One goal of the system designers should now be to address the parts of the interface that caused the most problems.

The Critical Review, although not a formally repeatable process, did provide some good insights into the quality of the interface. It is questionable whether it would be beneficial to repeat the Critical Review unless a large amount of new functionality is added to CU-SeeMe. Using empirical methods is much more rigorous and formal which makes it repeatable and more sound. However, we should point out that the Critical Review predicted 50% of the problems that users encountered during testing and 87% of the problems that were considered High importance according to user feedback. Some of the observations and predicted potential problems in the Critical Review apply to more than one observed problem in the Empirical Review, but these numbers do suggest a "poor man's" technique for improving interface design. This technique may be extendible to make it more methodical and, although it still may not be rigorous and scientifically solid, it should prove to be another useful tool in the Interface Designer's toolbox.

In the empirical testing, the design of the benchmark tasks was a simple and straightforward process. Instead of trying to analytically identify the parts of the system that need to be tested, we simply followed through the scenarios to distinguish what parts of the system would be used in the actual realization of each scenario. The scenarios define what the user might be doing and in what context, so we can then map this to a set of explicit instructions for carrying out these operations. This set of explicit instructions precisely shows one way to accomplish the benchmark task. There may be other ways to accomplish the goal, but since the experimental

designer is familiar with the software being tested, they should be able to map the *best* or most efficient way to complete the task. This can easily be turned into a set of user instructions by simply removing the *how* and letting the *what* remain. This method seemed to make the overall process of testing more clear to the participants since the tests "made sense" and could be linked to real world usage. If care and diligence are used in designing realistic and reasonable scenarios for software usage, then designing a set of empirical tests to measure the effectiveness of that software is a straightforward and natural task.

CU-SeeMe is by no means a trouble-ridden package with no hope but is a very exciting tool for most people who try it. Many individuals who use CU-SeeMe would not have the opportunity to experience video conferencing first hand if it weren't made affordable by CU-SeeMe. The ultimate goal of this study is to suggest a beginning to the process of making CU-SeeMe a better tool so that it will grow in popularity as time passes. The designers and programmers have made many successful improvements in the performance of CU-SeeMe. This Formative Evaluation is the first step in improving the user-centered effectiveness of CU-SeeMe.

6. References

[1] Argyle, M., Lalljee, M., and Cook, M. "The effects of visibility on interaction in a dyad." Human Relations, 21, 3-17. 1968.

[2] del Galdo, E.M., Williges, R.C., Williges, B.H., and Wixon, D.R. (1986) "An Evaluation of Critical Incidents for Software Documentation Design". Proceedings of Human Factors Society Conference, Anaheim, CA: Human Factors Society, 19-23.

[3] Carroll, J.M. & Rosson, M.B. (1985). "Usability Specifications as a Tool in Iterative Development." In H.R. Hartson (Ed.), Advances in Human-Computer Interaction, Vol. 1 (pp. 1-28). Norwood, NJ: Ablex.

[4] Elrod, Scott, Richard Bruce, Rich Gold, Daivid Goldberg, Frank Halasz, William Janssen, David Lee, Kim McCall, Elin Pederson, Ken Pier, John Tang, and Brent Welch. "LiveBoard: A Large Interactive Display Supporting Group Meetings, Presentations and Remote Collaboration." In proceedings of CHI, 1992 (Monterey, California, May 3-May 7, 1992) ACM, New York, 1992. pp. 599-607.

[5] Encyclopedia Brittanica, Vol 24. Encyclopedia Brittanica Inc, Chicago. 1992.

[6] Fish, Rober S., Robert E. Kraut, Robert W. Root. "Evaluating Video as a Technology for Informal Communication." In proceedings of CHI, 1992 (Monterey, California, May 3-May 7, 1992) ACM, New York, 1992. pp. 37-48.

[7] Flanagan, J.C.(1954) "The Critical Incident Technique". Psychological Bulletin, 51(28), 28-35.

[8] Gaver, William, Thomas Moran, Allan MacLean, Lennart Lövstrand, Paul Dourish, Kathleen Carter, William Buxton. "Realizing A Video Environment: EuroPARC's RAVE System." In proceedings of CHI, 1992 (Monterey, California, May 3-May 7, 1992) ACM, New York, 1992. pp. 27-35.

[9] Hix, D., and Hartson, H. Rex. Developing User Interfaces: Ensuring Usability Through Product & Process. John Wiley and Sons. New York. 1993.

[10] Hollan, Jim, and Scott Stornetta. "Beyond Being There." In proceedings of CHI, 1992 (Monterey, California, May 3-May 7, 1992) ACM, New York, 1992. pp. 119-125.

[11] Ishii, Hiroshi, and Minoru Kobayashi. "ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact." In proceeding of CHI, 1992 (Monterey, California, May 3-May 7, 1992) ACM, New York, 1992. pp. 525-532.

[12] Kendon, A. & Ferber, A. "A Description of Some Human Greetings." In R. Michael & J. Crook (Eds.), Comparitive Ecology and Behaviour of Primates. London: Academic Press.

[13] Koenemann-Belliveau, Jürgen, John M. Carroll, Mary Beth Rosson, and Mark K. Singley. "Comparative Usability Evaluation: Critical Incidents and Critical Threads". In proceedings of CHI 1994 (Boston, MA, April 24-April 28, 1994) ACM: New York, 1994. pp. 245-251.

[14] Labriola, Don, "Meeting on the Edge", Windows Sources Vol ii, no. 9. Sept 1994 pp97-116.

[15] Murray, Janet, "K12 Network: Global Education through Telecommunications", Communications of the ACM, v36. no.8, August 1993.

[16] Nielson, J. and Molich R. (1990). "Heuristic Evaluation of User Interfaces". Proceedings of CHI Conference on Human Factors in Computing Systems, New York: ACM, 249-256.

[17] Parker, Tracy LaQuey, "The Internet and Schools: A Survey of Networking Activities", presented at INET '94 conference in Prague.

[18] Parker, Tracy LaQuey, "The Internet-K12 Connection: How Students and Teachers are Using the Internet", The

Interoperability Report. Foster City, CA.: Interop, Inc. April 1994.

[19] Schneider, M.L. "Models for the Design of Static Software User Assistance." Directions in Human/Computer Interaction (Eds. Albert Badre, Ben Schneiderman). Ablex Publishing. Norwood, New Jersey. 1982.

[20] Schneiderman, Ben. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley. Reading, Mass. 1987.

[21] Sellen, Abigail J., "Speech Patterns in Video-Mediated Conversations." In proceedings of CHI, 1992 (Monterey, California, May 3-May 7, 1992) ACM, New York, 1992. pp. 49-59.

[22] Short, J., Williams, E., and Christie, B. The Social Psychology of Telecommuncations. London: Wiley & Sons. 1976.

[23] Smith, S.L. and Mosier, J.N. (1986). "Guidelines for Designing User Interface Software". (ESD-TR-86-278/MTR 10090). Bedford, MA: MITRE Corporation. On BRUITΔSAM Hypercard® Stack for Macintosh®.

[24] Whiteside, J., Bennet, J., and Holtzblatt, K. (1988). "Usability Engineering: Our Experience and Evolution". In M. Helander (Ed.), Handbook of Human-Computer Interaction pp. 791-817. Amsterdam: Elsevier North-Holland.

7. Appendix A- Critical Review Guidelines

This subset of guidelines comes from Hix, Hartson [9] in chapters 2 and 3. The full Smith and Mosier [23] set of 944 guidelines is too large for the purposes of this review and although we can gather from Smith and Mosier a relevant subset of guidelines, Hix and Hartson already present a useful set of general guidelines which are taken from Smith and Mosier. Please reference either the Smith and Mosier Guidelines or Hix, Hartson for an explanation of each guideline.

GENERAL

1. Prevent user errors
2. Optimize user operations
3. Keep locus of control with user
4. Help user get started with system
5. Give user a mental model based on user tasks
6. Be consistent
7. Keep it simple
8. Account for human memory limitations by giving the user frequent closure on tasks
9. Use recognition rather than recall
10. Use cognitive directness (minimize mental transformations)
11. Draw on real world analogies
12. Use informative feedback
13. Give the user appropriate status indicators
14. Use user-centered wording in messages
15. Use positive, non-threatening wording
16. Use specific, constructive terms in error messages
17. Do not anthropomorphize
18. Use modes cautiously
19. Make user actions easily reversible
20. Get users' attention judiciously
21. Maintain display inertia
22. Organize the screen to manage complexity
23. Accomodate user experience levels

WINDOWS

1. Don't overuse windows (minimize window manipulation)
2. Appearand behavior of the primary window should be consistent
3. Use different windows for different independent tasks
4. Use different windows for different coordinated views of the same task

MENUS

1. Use meaningful groupings of menu choices
2. Use meaningful ordering of menu choices
3. Use brief descriptions for menu choices
4. Use a consistent layout across all menus, and keep the screen uncluttered
5. Allow shortcuts

FORMS

1. Use consistent, visually appealing layout and content
2. Use appropriate visual cues for fields on forms
3. Use local navigation among fields
4. Use local navigation within fields

GRAPHICAL INTERFACE

1. Use real world analogies as much as possible
2. Show different views of the same visual object
3. Keep the visual representation as simple as possible

8. Appendix B - Instructions and Protocol

8.1 Experimental Protocol

The purpose of this experiment is to evaluate and enhance the quality of the user interface for the CU-SeeMe video-conferencing package from Cornell University. An empirical formative evaluation will be performed on a single, existing interface design using five human test participants. Both quantitative and qualitative data will be collected on each participant. Quantitative data will be collected through timing task completions, recording error rates, and through scaled questionnaires. Qualitative data will be collected through verbal exchange between the participants and experimenters at the completion of each task.

There will be several benchmark tasks that each participant will be asked to perform that will be broken into six task sets. The tasks all follow a likely set of scenarios that describe how this system might be used in a real world setting. Each participant will receive an informed consent form before beginning the experiments along with a verbal introduction from the experimenters of what is expected. Either a second experimenter or an automated timing and counting package such as Ideal will be used with each participant. After each of the six task sets, the participants will be asked to complete a short questionnaire and then asked to remark on problems they had, expectations that were not met, or suggestions for improvement. Each session will be videotaped for later review of the participants' reactions during the tasks and will last approximately one hour. The participants will be selected from the pool of teachers in the local elementary school system or from the staff at Virginia Tech, on a strictly volunteer basis. The criteria for selection is that each participant be able to relate the system to an educational use, meaning that each participant be either a teacher or a student. All participants will be at least eighteen years of age. Novice to intermediate computer skill is preferred, and expert users will be kept to a minimum as participants.

The data gathered will be incorporated into a written report that will analyze the current interface and provide insight into the proper direction to take as the interface evolves. A critical review based on random inputs, the experimenters experience, and Human-Computer-Interaction knowledge will accompany the empirical results.

There are no known risks involved in this experiment and the data will be held confidential with access limited to Michael Bibeau and Dr. Roger Ehrich.

8.2 Participant Instructions

Thank you for volunteering to help in this experiment.

Please read through the instructions completely before beginning the experiment.

CU-SeeMe is a video-conferencing software package from Cornell University that is currently public-domain. The current version of the software is still a Beta-version meaning that it is not yet fully developed. Through this experiment, we hope to provide useful input into the interface design of CU-SeeMe and to identify some important features for a video-conferencing package to be used in an educational setting.

CU-SeeMe combines live video and audio into a series of conference windows. Each participant in a video-conference has their own window that displays their video image. The name of each participant appears over the top of their respective window. A video-conference can consist of two or more parties. A two party conference is conducted by the two computers linking directly to each other. A multiparty conference is conducted through the use of a third computer, called a *reflector.* In a multiparty conference, each participant links their computer to the reflector and the reflector re-transmitts every signal to everyone else connected.

An individual connected to a conference does not necessarily have to send and receive both video and audio signals from the other parties in the conference. Three people might have a conference where two of the parties can see and hear each other, and the third party can see and hear both of them, but they cannot see nor hear him/her. Various configurations are possible with CU-SeeMe; from a full blown conference with several people talking to, and seeing each other, down to a simple one-way conversation where two people are connected with audio-only.

In this experiment, you will be asked to perform six task sets. Please read each instruction sheet completely and clear up any questions before beginning each set of tasks. Tell the experimenter when you are ready to begin and he will tell you to start. If you need help during the experiment, please consult the documentation provided.

Please do not become embarrassed if a task does not make sense. We are here to evaluate the quality of the current interface and find ways to improve the interface. We are NOT concerned with testing your personal abilities. If you are confused, agitated, or have comments, please share them with the experimenter at the completion of each task set. Also, we are interested in the quality of the documentation that we have provided. If you find it helpful, please be sure to let the experimenter know.

Again, thank you for participating in this testing. The feedback that you provide will be used to develop better software interfaces that you can benefit from in the future.

8.2.1 Task 1

PLEASE READ ENTIRELY AND DO EACH STEP BEFORE MOVING ON. IF YOU GET STUCK, TRY THE MANUAL. THE EXPERIMENTOR MAY PROMPT YOU IF IT LOOKS LIKE YOU CANNOT FIND THE INFORMATION YOU NEED TO COMPLETE THE TASK.

In task 1 you will be operating CU-SeeMe in a minimal configuration. CU-SeeMe is designed to work even without a camera or a microphone.

The goal of this task set is to start the program, and make a connection. You will try to communicate information through CU-SeeMe without the use of a camera or audio microphone by typing in the video window.

1) Start CU-SeeMe by double-clicking on the CU-SeeMe icon.

2) Start a conference with Mr. Math at the following IP address: _____ Set the connection so that you will both Send and Receive video.

3) When Mr. Math's window appears and he says "Hello", *type* the message "Hello, heres my IP address:_____"

4) When you confirm that he has your IP address, ask when your next meeting should be and write it down.

5) Ask Mr. Math for his IP address. (Pretend you lost it)

6) When you have his IP address, add Mr. Math to your Nickname list so next time you don't have to remember the address. Set it up so that you will automatically send and receive when you connect to Mr. Math using his Nickname.

7) Disconnect the conference and exit CU-SeeMe.

8.2.2 Task 2

PLEASE READ ENTIRELY AND DO EACH STEP BEFORE MOVING ON. IF YOU GET STUCK, TRY THE MANUAL. THE EXPERIMENTOR MAY PROMPT YOU IF IT LOOKS LIKE YOU CANNOT FIND THE INFORMATION YOU NEED TO COMPLETE THE TASK.

In task 2 you will conduct your second conference with Mr. Math. This time you have a camera and microphone. You will configure your camera and audio so that Mr. Math can see you and hear you during the conference.

1) Start CU-SeeMe by double clicking on the CU-SeeMe icon.

2) Adjust the video camera so that you are centered in the window.

3) Adjust the brightness and contrast of your video to your desired level.

4) Set your audio to Δ-mod(16kb/s).

5) Start a conference with Mr. Math and ensure the audio window is visible.

6) Use audio to convey the following message to Mr. Math. "Lets have your students do show-and-tell on Wednesday. What time is good for you?"

7) When Mr. Math confirms the day and gives you a time, confirm the time and then say "Goodbye".

8) Disconnect the conference and close Mr. Math's window.

9) Close the settings box and the information line on your local window.

8.2.3 Task 3

PLEASE READ ENTIRELY AND DO EACH STEP BEFORE MOVING ON. IF YOU GET STUCK, TRY THE MANUAL. THE EXPERIMENTOR MAY PROMPT YOU IF IT LOOKS LIKE YOU CANNOT FIND THE INFORMATION YOU NEED TO COMPLETE THE TASK.

In task 3, you will be configuring some of the other parts of CU-SeeMe.

1) Make a connection to yourself so that you now have two windows, both with your picture.

2) Arrange the two windows side by side.

3) Save the current window positions.

4) Set the resolution of your picture to Standard Resolution.

5) Set transmission parameters as: Min. Kbps/sec to 20, Max. Kbps/sec to 90, and the Max Frame Rate to 25.

6) Disconnect from yourself.

7) Close the second window.

8.2.4 Task 4

PLEASE READ ENTIRELY AND DO EACH STEP BEFORE MOVING ON. IF YOU GET STUCK, TRY THE MANUAL. THE EXPERIMENTOR MAY PROMPT YOU IF IT LOOKS LIKE YOU CANNOT FIND THE INFORMATION YOU NEED TO COMPLETE THE TASK.

In task 4 you will conduct a multi-party conference where all three parties have full video and two have audio. Mr. English cannot transmit audio.

1) Establish a Connection with the Info Reflector. Ensure both Send and Receive are enabled.

2) Arrange the three conference windows so they are all in a row.

3) Establish a private talk channel with Mr. Math and ask "Can anyone hear me?"

4) Ensure that Mr. Math can hear you and that Mr. English cannot by listening for Mr. Math to respond "Yes, Mr. Math heard you".

5) Once Mr. Math has responded, type "Did you hear me Mr. English?" in your video window. (Mr. English should respond by typing "No".)

6) Close the private talk channel with Mr. Math.

7) Now, disable Mr. Math's audio so that you cannot hear him.

8) Ask Mr. Math to say "Hello" and wave to you.

9) Ensure that you cannot hear Mr. Math and then ask Mr. English if he heard him. (Mr. English should type back "yes")

10) Enable Mr. Math's audio.

11) Stop Sending your video picture but remain connected to the conference. The eye on the other two windows will close, indicating they cannot see you.

12) Close your local video window and close Mr. Math's window.

13) Enlarge Mr. English's window and move it to the center of the screen.

14) Close the audio window.

8.2.5 Task 5

PLEASE READ ENTIRELY AND DO EACH STEP BEFORE MOVING ON. IF YOU GET STUCK, TRY THE MANUAL. THE EXPERIMENTOR MAY PROMPT YOU IF IT LOOKS LIKE YOU CANNOT FIND THE INFORMATION YOU NEED TO COMPLETE THE TASK.

In task 5 you will be resuming a normal conference after Mr. English from task 4 has finished his [imaginary] talk.

1) Resume Sending your video. The eye on Mr. English's window should open.

2) Open your local window.

3) Reopen Mr. Math's window.

4) Open the audio window.

5) Ask Mr. Math if he can hear you and wait for him to reply "Yes".

6) Tell Mr. English that one of your students has a question and wait for him to type back "Go ahead".

7) Ask Mr. English "What kind of tea do you prefer at tea time?" and wait for him to type the answer.

8) Say "Thank You" and "Goodbye" to Mr. English and Mr. Math. When both have replied "Goodbye", disconnect from the conference.

9) Close Mr. English's window.

10) Close Mr. Math's window.

11) Delete Mr. Math from your Nickname list.

8.2.6 Task 6 - Free Play

Take the remainder of the time to explore CU-SeeMe. You will be provided with a list of Reflector sites that you can connect to and try to find new people to talk with. The experimenter will stop you after about 10 minutes.

1) Play around with the system for the remainder of the experiment until the experimenter asks you to stop. Try connecting to some sites and see what you can find.

2) Try the Cornell Reflector at least once. There is usually activity on that reflector.

3) Disconnect and exit CU-SeeMe when you are finished.

8.3 Questionnaires[17]

[17] Adapted from: Schniederman, Ben, Designing the User Interface: Strategies for Effective Human Computer Interaction. Addison Wesley, Reading Mass. pp.398-407, 1987.

8.3.1 Questionnaire #1

Please answer the following questions as they relate to task #1.

Circle the number on the scale that most closely matches your reaction to the operations just completed.

1) Terminology relates to task domain: distant ... closely
0 1 2 3 4 5 6 7 8 9 10 NA

2) Operations relate to tasks: distantly ... closely
0 1 2 3 4 5 6 7 8 9 10 NA

3) Number of operations per task: many ... few
0 1 2 3 4 5 6 7 8 9 10 NA

4) Informative feedback provided: never ... always
0 1 2 3 4 5 6 7 8 9 10 NA

5) Amount of feedback: too little ... adequate
0 1 2 3 4 5 6 7 8 9 10 NA

6) Display layouts simplify tasks: never ... always
0 1 2 3 4 5 6 7 8 9 10 NA

7) Displays: cluttered ... uncluttered
0 1 2 3 4 5 6 7 8 9 10 NA

8) Sequence of displays: confusing ... clear
0 1 2 3 4 5 6 7 8 9 10 NA

9) Next screen in a sequence: unpredictable ... predictable
0 1 2 3 4 5 6 7 8 9 10 NA

10) Error Correction: confusing ... clear
0 1 2 3 4 5 6 7 8 9 10 NA

11) Learning the operation: difficult ... easy
0 1 2 3 4 5 6 7 8 9 10 NA

12) Getting started: difficult ... easy
0 1 2 3 4 5 6 7 8 9 10 NA

13) Supplemental Reference Materials: confusing ... clear
0 1 2 3 4 5 6 7 8 9 10 NA

14) Overall Reactions: uninteresting ... interesting
0 1 2 3 4 5 6 7 8 9 10 NA

dull ... stimulating
0 1 2 3 4 5 6 7 8 9 10 NA

8.3.2 Questionnaire #2

Please answer the following questions as they relate to task #2.

Circle the number on the scale that most closely matches your reaction to the operations just completed.

1) Operations relate to tasks:
distantly closely
0 1 2 3 4 5 6 7 8 9 10 NA

2) Number of operations per task:
many few
0 1 2 3 4 5 6 7 8 9 10 NA

3) Informative feedback provided:
never always
0 1 2 3 4 5 6 7 8 9 10 NA

4) Amount of feedback:
too little adequate
0 1 2 3 4 5 6 7 8 9 10 NA

5) Display layouts simplify tasks:
never always
0 1 2 3 4 5 6 7 8 9 10 NA

6) Displays:
cluttered uncluttered
0 1 2 3 4 5 6 7 8 9 10 NA

7) Sequence of displays:
confusing clear
0 1 2 3 4 5 6 7 8 9 10 NA

8) Next screen in a sequence:
unpredictable predictable
0 1 2 3 4 5 6 7 8 9 10 NA

9) Error messages are helpful:
never always
0 1 2 3 4 5 6 7 8 9 10 NA

10) Error correction:
confusing clear
0 1 2 3 4 5 6 7 8 9 10 NA

11) Learning the operation:
difficult easy
0 1 2 3 4 5 6 7 8 9 10 NA

12) Learning more features:
difficult easy
0 1 2 3 4 5 6 7 8 9 10 NA

13) Supplemental Reference Materials:
confusing clear
0 1 2 3 4 5 6 7 8 9 10 NA

14) Overall Reactions:
frustrating satisfying
0 1 2 3 4 5 6 7 8 9 10 NA

difficult easy
0 1 2 3 4 5 6 7 8 9 10 NA

8.3.3 Questionnaire #3

Please answer the following questions as they relate to task #3.

Circle the number on the scale that most closely matches your reaction to the operations just completed.

1) Terminology relates to task domain: distant closely
0 1 2 3 4 5 6 7 8 9 10 NA

2) Display layouts simplify tasks: never always
0 1 2 3 4 5 6 7 8 9 10 NA

3) Displays: disorderly orderly
0 1 2 3 4 5 6 7 8 9 10 NA

4) Sequence of displays: confusing clear
0 1 2 3 4 5 6 7 8 9 10 NA

5) Next screen in a sequence: unpredictable predictable
0 1 2 3 4 5 6 7 8 9 10 NA

6) Learning the operation: difficult easy
0 1 2 3 4 5 6 7 8 9 10 NA

7) Learning more features: difficult easy
0 1 2 3 4 5 6 7 8 9 10 NA

8) Exploration of features: discouraged encouraged
0 1 2 3 4 5 6 7 8 9 10 NA

9) Supplemental Reference Materials: confusing clear
0 1 2 3 4 5 6 7 8 9 10 NA

10) Overall Reactions: frustrating satisfying
0 1 2 3 4 5 6 7 8 9 10 NA

difficult easy
0 1 2 3 4 5 6 7 8 9 10 NA

8.3.4 Questionnaire #4

Please answer the following questions as they relate to task #4.

Circle the number on the scale that most closely matches your reaction to the operations just completed.

1) Terminology relates to task domain: distant ... closely
0 1 2 3 4 5 6 7 8 9 10 NA

2) Informative feedback provided: never ... always
0 1 2 3 4 5 6 7 8 9 10 NA

3) Display layouts simplify tasks: never ... always
0 1 2 3 4 5 6 7 8 9 10 NA

4) Displays: cluttered ... uncluttered
0 1 2 3 4 5 6 7 8 9 10 NA

5) Sequence of displays: confusing ... clear
0 1 2 3 4 5 6 7 8 9 10 NA

6) Next screen in a sequence: unpredictable ... predictable
0 1 2 3 4 5 6 7 8 9 10 NA

7) Learning the operation: difficult ... easy
0 1 2 3 4 5 6 7 8 9 10 NA

8) Learning more features: difficult ... easy
0 1 2 3 4 5 6 7 8 9 10 NA

9) Exploration of features: discouraged ... encouraged
0 1 2 3 4 5 6 7 8 9 10 NA

10) Supplemental Reference Materials: confusing ... clear
0 1 2 3 4 5 6 7 8 9 10 NA

11) Overall Reactions: uninteresting ... interesting
0 1 2 3 4 5 6 7 8 9 10 NA

dull ... stimulating
0 1 2 3 4 5 6 7 8 9 10 NA

8.3.5 Questionnaire #5

Please answer the following questions as they relate to task #5.

Circle the number on the scale that most closely matches your reaction to the operations just completed.

1) Terminology relates to task domain: distant ... closely
0 1 2 3 4 5 6 7 8 9 10 NA

2) Operations relate to tasks: distantly ... closely
0 1 2 3 4 5 6 7 8 9 10 NA

3) Informative feedback provided: never ... always
0 1 2 3 4 5 6 7 8 9 10 NA

4) Display layouts simplify tasks: never ... always
0 1 2 3 4 5 6 7 8 9 10 NA

5) Displays: cluttered ... uncluttered
0 1 2 3 4 5 6 7 8 9 10 NA

6) Sequence of displays: confusing ... clear
0 1 2 3 4 5 6 7 8 9 10 NA

7) Next screen in a sequence: unpredictable ... predictable
0 1 2 3 4 5 6 7 8 9 10 NA

8) Learning the operation: difficult ... easy
0 1 2 3 4 5 6 7 8 9 10 NA

9) Learning more features: difficult ... easy
0 1 2 3 4 5 6 7 8 9 10 NA

10) Exploration of features: discouraged ... encouraged
0 1 2 3 4 5 6 7 8 9 10 NA

11) Supplemental Reference Materials: confusing ... clear
0 1 2 3 4 5 6 7 8 9 10 NA

12) Overall Reactions: frustrating ... satisfying
0 1 2 3 4 5 6 7 8 9 10 NA

difficult ... easy
0 1 2 3 4 5 6 7 8 9 10 NA

8.3.6 Task Interview Form

This is a standardized form for interviewing each participant between task sets. Instead of a single interview at the end of the entire experiment, I have chosen to conduct shorter interviews between individual tasks when problems, thoughts, or suggestions are still clear in the participant's head.

1) Were there any specific items in this task set that gave you particular trouble?

If yes, what were they and do you have suggestions to fix the problems?

2) Were there any specific items in this task set that you felt were straightforward and simple to carry out?

If yes, what were they?

3) While performing the task set, were there any capabilities of the system that you expected to see that were not present?

If yes, what were they?

4) Was the documentation provided helpful when you needed it?

If no, what changes might make it more helpful?

8.3.7 Final Interview Question

Do you see any potential uses for this system that may require different or additional functionality to be added to the interface?

Other Comments or suggestions at this point?

8.3.8 Demographics Form

Participant #_____

Occupation_____

What is your experience with computers?

__use daily

__use once a week

__use once a month

__use less than once a month

__never use

What is your experience with the Internet?

__use daily

__use once a week

__use once a month

__use less than once a month

__never use

What is your experience with the Macintosh?

__use regularly

__use infrequently

__never use

What is your experience using a mouse?

__use regularly

__use infrequently

__never use

If you do have Internet experience, what type of Internet applications have you used? (check all that apply)

__Web Browsers. (NetScape, Mosaic, Cello, WinWeb, etc..)

__E-mail. (Eudora, ELM, other e-mail packages)

__Gopher.

__FTP.

__Telnet.

Have you ever used CU-SeeMe? __yes __no

If yes, how often?

__1-2 times __3-5 times __6-10 times __11 or more times

Have you ever done any type of videoconferencing other than CU-SeeMe? __yes __no
If yes, was it computer-based? __yes __no
If computer-based, what was the name of the software?_____

8.3.9 Informed Consent

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

Informed Consent for Participants
of Investigative Projects

Title of Project: **A Formative Interface Evaluation of CU-SeeMe**
Principal Investigator: **Michael J. Bibeau**

I. THE PURPOSE OF THIS RESEARCH

You are invited to participate in a study about Human-Computer-Interaction. This study involves experimentation for the purpose of evaluating the interface for CU-SeeMe.

II. PROCEDURES

The procedures to be used in this research are as follows: You will be given a set of instructions, each containing a series of small tasks that all relate to one, overall task. You will be asked to complete the tasks using CU-SeeMe while being observed. After each set of tasks, you will be asked to fill out a small questionnaire and then engage in short verbal exchange with the experimenter. The time and conditions required for you to participate in this project are: You must be at least eighteen years of age and have had some experience as either a teacher or a student. The time required to complete this experiment will be approximately one hour.

The possible risks or discomfort to you as a participant are none relating to the experiment.

III. BENEFITS OF THIS PROJECT

Your participation in the project will provide insight into better user interfaces for desktop videoconferencing systems.

No guarantee of benefits has been made to encourage you to participate.

You may receive a synopsis or summary of this research when completed. Please leave a self-addressed envelope or other appropriate means for you to receive the information when it is available.

IV. EXTENT OF ANONYMITY AND CONFIDENTIALITY

The results of this study will be kept strictly confidential. At no time will the researchers release the results of the study to anyone other than individuals working on the project without your written consent. The information you provide will have your name removed and only a participant number will identify you during analysis and any written reports of the research.

The experiment will be video-taped. These tapes will only be reviewed by Michael Bibeau and will be erased after March 31, 1995.

V. COMPENSATION

There is no monetary compensation offered for participation in this study.

VI. FREEDOM TO WITHDRAW

You are free to withdraw from this study at any time without penalty.

VII. APPROVAL OF RESEARCH

This research project has been approved, as required, by the Institutional Review Board for projects involving human subjects at Virginia Polytechnic Institute and State University, by the Department of Computer Science.

VIII. SUBJECT'S RESPONSIBILITIES

I know of no reason I cannot participate in this study.

_____

Signature

----------**detach here**----------

IX. SUBJECT'S PERMISSION

I have read and understand the informed consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project.

If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project.

Should I have any questions about this research or its conduct, I will contact:

Michael J. Bibeau Investigator	**951-2731**
Dr. R.E. Ehrich Faculty Advisor	
Ernest R. Stout Chair, IRB Research Division	**1x9359**

9. Appendix C - Complete Questionnaire Breakdown

9.1 Participant 1

Question		Participant #1				
Task:		**1**	**2**	**3**	**4**	**5**
1) Terminology relates to task domain:	(distant/closely)	3		2	5	4
2) Operations relate to tasks:	(distant/closely)	4	6			5
3) Number of operations per task:	(many/few)	3	5			
4) Informative feedback provided:	(never/always)	5	6		6	5
5) Amount of feedback:	(too little/adequate)	6	6			
6) Display layouts simplify tasks:	(never/always)	3	5	5	5	4
7) Displays:	(cluttered/uncluttered)	4	5	3	5	4
8) Sequence of displays:	(confusing/clear)	3	5	4	6	4
9) Next screen in a sequence:	(unpredictable/predictable)	3	5	4	5	4
10) Error messages are helpful:	(never/always)		4			
11) Error correction:	(confusing/clear)	2	4			
12) Learning the operation:	(difficult/easy)	2	4	5	5	4
13) Learning more features:	(difficult/easy)		5	5	6	5
14) Exploration of features:	(discouraged/encouraged)			5	6	5
15) Getting started:	(difficult/easy)	2				
16) Supplemental reference materials:	(confusing/clear)	2	3			
17) Overall reactions:	(uninteresting/interesting)	7			7	
	(dull/stimulating)	8			6	
	(frustrating/satisfying)		4	4		4
	(difficult/easy)		4	4		5
Mean		3.80	4.73	4.10	5.64	4.42
St. D.		1.90	0.88	0.99	0.67	0.51
Med.		3.00	5.00	4.00	6.00	4.00

9.2 Participant 2

Question		Participant #2 Task: 1	2	3	4	5
1) Terminology relates to task domain:	(distant/closely)	5				
2) Operations relate to tasks:	(distant/closely)	5	10			
3) Number of operations per task:	(many/few)	5	3			
4) Informative feedback provided:	(never/always)	4	1			
5) Amount of feedback:	(too little/adequate)	1	1			
6) Display layouts simplify tasks:	(never/always)	4	1			
7) Displays:	(cluttered/uncluttered)	8	10			
8) Sequence of displays:	(confusing/clear)	6	1			
9) Next screen in a sequence:	(unpredictable/predictable)		1			
10) Error messages are helpful:	(never/always)		1			
11) Error correction:	(confusing/clear)	1	0			
12) Learning the operation:	(difficult/easy)	1	0			
13) Learning more features:	(difficult/easy)		0			
14) Exploration of features:	(discouraged/encouraged)					
15) Getting started:	(difficult/easy)	1				
16) Supplemental reference materials:	(confusing/clear)	10	5			
17) Overall reactions:	(uninteresting/interesting)	7				
	(dull/stimulating)	7				
	(frustrating/satisfying)		0			
	(difficult/easy)		0			
Mean		4.64	2.27			
St. D.		2.87	3.41			
Med.		5.00	1.00			

9.3 Participant 3

Question		Participant #3				
Task:		1	2	3	4	5
1) Terminology relates to task domain:	(distant/closely)	8		7	8	8
2) Operations relate to tasks:	(distant/closely)	8	9			8
3) Number of operations per task:	(many/few)	9	6			
4) Informative feedback provided:	(never/always)	8	7		6	6
5) Amount of feedback:	(too little/adequate)	8	7			
6) Display layouts simplify tasks:	(never/always)	8	8	5	6	7
7) Displays:	(cluttered/uncluttered)	8	9	8	8	9
8) Sequence of displays:	(confusing/clear)	8	8	7	8	8
9) Next screen in a sequence:	(unpredictable/predictable)	6	8	8	8	7
10) Error messages are helpful:	(never/always)		5			
11) Error correction:	(confusing/clear)	7	5			
12) Learning the operation:	(difficult/easy)	5	7	5	4	5
13) Learning more features:	(difficult/easy)		7	5	6	5
14) Exploration of features:	(discouraged/encouraged)			8	8	8
15) Getting started:	(difficult/easy)	2				
16) Supplemental reference materials:	(confusing/clear)	8	5	8	7	
17) Overall reactions:	(uninteresting/interesting)	8			8	
	(dull/stimulating)	8			8	
	(frustrating/satisfying)		5	8		8
	(difficult/easy)		5	5		6
Mean		7.27	6.73	6.73	7.08	7.08
St. D.		1.75	1.49	1.42	1.31	1.31
Med.		8.00	7.00	7.00	8.00	7.50

9.4 Participant 4

Question		Participant #4				
Task:		1	2	3	4	5
1) Terminology relates to task domain:	(distant/closely)	10		10	9	10
2) Operations relate to tasks:	(distant/closely)	10	9			10
3) Number of operations per task:	(many/few)	9	10			
4) Informative feedback provided:	(never/always)	10	8		9	10
5) Amount of feedback:	(too little/adequate)	10	9			
6) Display layouts simplify tasks:	(never/always)	10	7	9	9	10
7) Displays:	(cluttered/uncluttered)	9	10	10	10	10
8) Sequence of displays:	(confusing/clear)		8	9	9	10
9) Next screen in a sequence:	(unpredictable/predictable)	9	8	9	9	10
10) Error messages are helpful:	(never/always)					
11) Error correction:	(confusing/clear)	10	5			
12) Learning the operation:	(difficult/easy)	10	6	10	9	10
13) Learning more features:	(difficult/easy)		7	10	9	10
14) Exploration of features:	(discouraged/encouraged)				9	
15) Getting started:	(difficult/easy)	10				
16) Supplemental reference materials:	(confusing/clear)	10	7	10	8	10
17) Overall reactions:	(uninteresting/interesting)	10			9	
	(dull/stimulating)	10			9	
	(frustrating/satisfying)		8	10		10
	(difficult/easy)		8	10		10
Mean		9.79	7.86	9.70	9.00	10.00
St. D.		0.43	1.41	0.48	0.43	0.00
Med.		10.00	8.00	10.00	9.00	10.00

9.5 Participant 5

Question		Participant #5				
Task:		1	2	3	4	5
1) Terminology relates to task domain:	(distant/closely)	8		9	9	8
2) Operations relate to tasks:	(distant/closely)	4	8			8
3) Number of operations per task:	(many/few)	8	8			
4) Informative feedback provided:	(never/always)	1	1		3	3
5) Amount of feedback:	(too little/adequate)	1	1			
6) Display layouts simplify tasks:	(never/always)	5	2	3	4	3
7) Displays:	(cluttered/uncluttered)	9	8	6	9	9
8) Sequence of displays:	(confusing/clear)	4	2	9	6	10
9) Next screen in a sequence:	(unpredictable/predictable)	8	9	9	9	9
10) Error messages are helpful:	(never/always)		1			
11) Error correction:	(confusing/clear)	6	3			
12) Learning the operation:	(difficult/easy)	7	7	9	8	7
13) Learning more features:	(difficult/easy)		0	3	7	9
14) Exploration of features:	(discouraged/encouraged)			5	4	4
15) Getting started:	(difficult/easy)	1				
16) Supplemental reference materials:	(confusing/clear)					
17) Overall reactions:	(uninteresting/interesting)	4			7	
	(dull/stimulating)	4			8	
	(frustrating/satisfying)		4	7		9
	(difficult/easy)		4	8		8
Mean		5.00	4.14	6.80	6.73	7.25
St. D.		2.77	3.21	2.44	2.20	2.49
Med.		4.50	3.50	7.50	7.00	8.00

10. Appendix D - CU-SeeMe Interface Redesign

This final section summarizes some of the solutions suggested in both the Analytical and Empirical portions of the evaluation as one possible redesign of some portions of the CU-SeeMe interface. Refer to the CU-SeeMe Users Manual for screen shots of the current interface design.

1. Provide a well constructed User's Manual with a How To section.
2. Keep the startup screen constant and disable items unavailable due to missing hardware. Inform the user what is missing on startup if the configuration is not complete.
3. When a Connection is established from the Nickname list, simply open the connection. If the user turns on the *Confirm Connections* in the *Conference Settings* then display the *Connect Window* before connecting.
4. Confirm operations; *Disconnect*, *Stop Sending*, and *Exit*. Give the ability to turn this off in *Settings* along with the ability to turn on confirmation for other operations that change the system state.
5. Give descriptive error messages with possible solutions when feasible. (especially on startup)

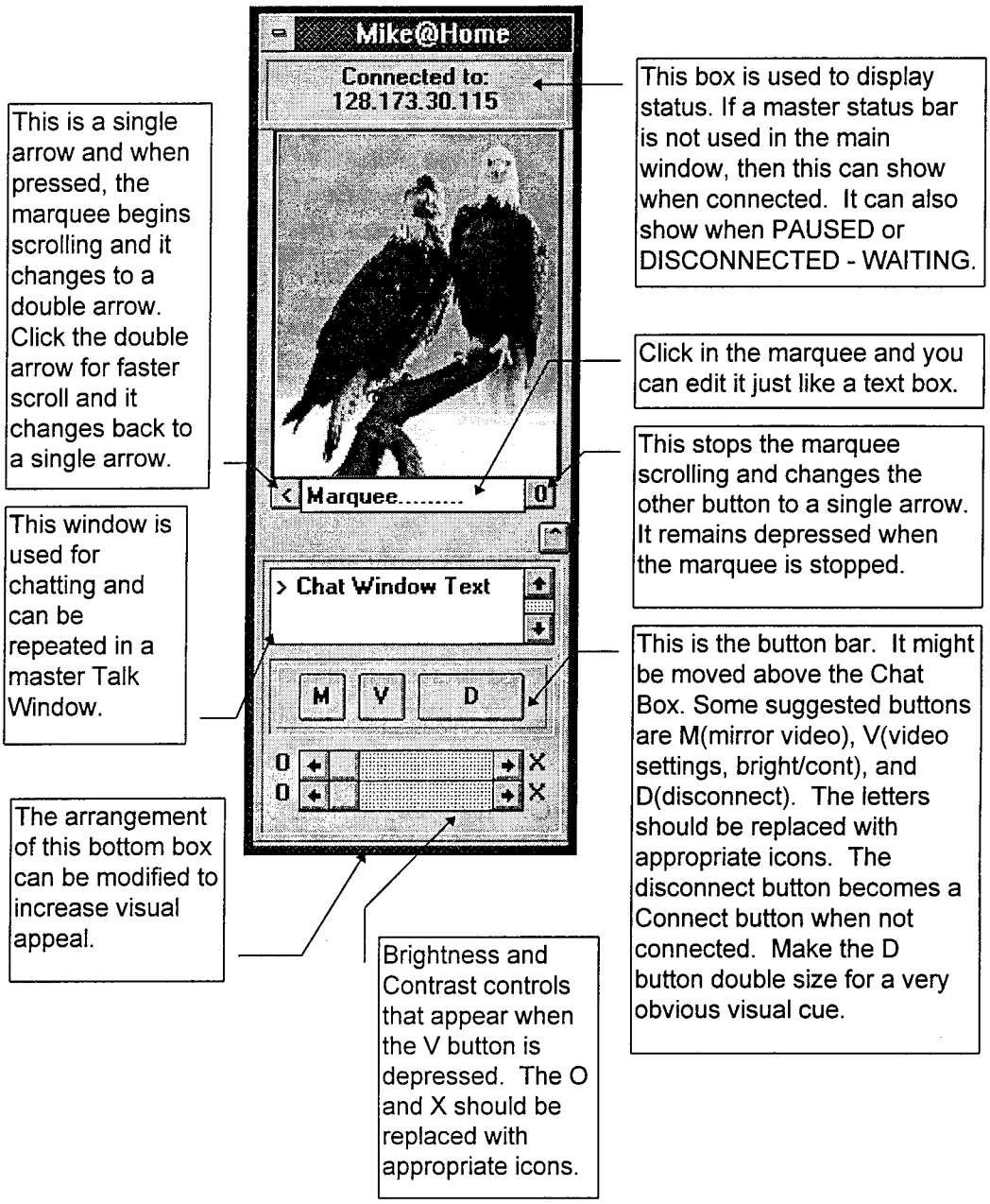


Figure 4: Redesigned Local Window

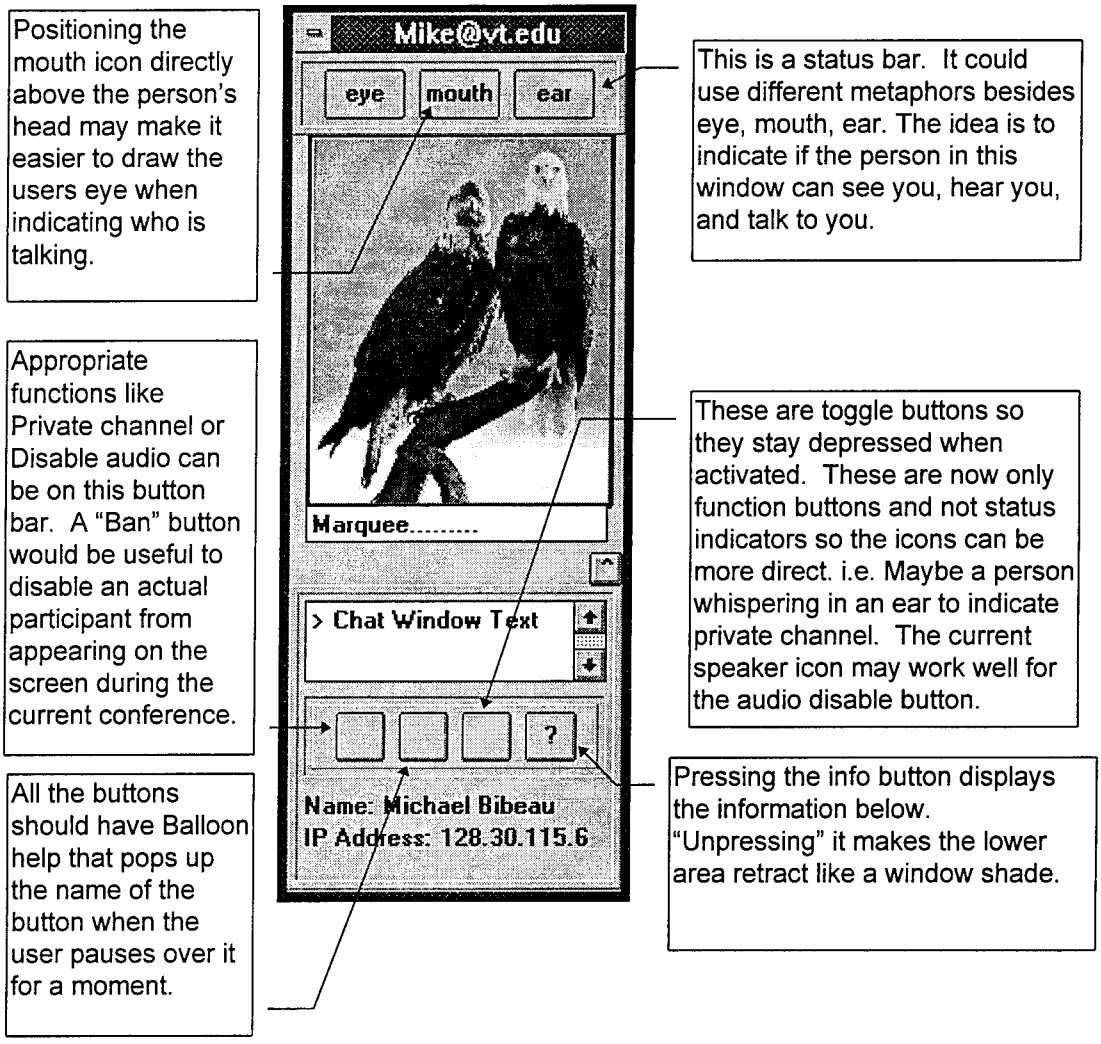


Figure 5: Redesigned Second Party Window

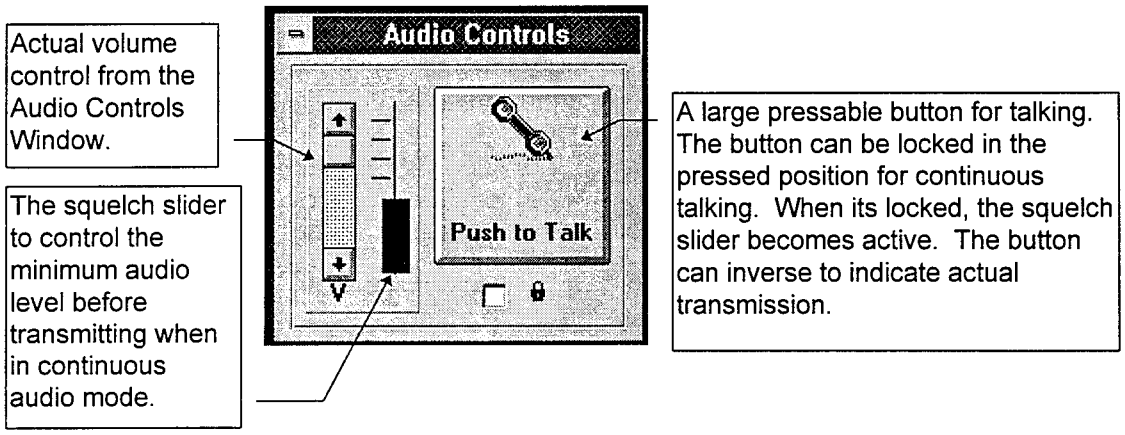


Figure 6: Redesigned Audio Window

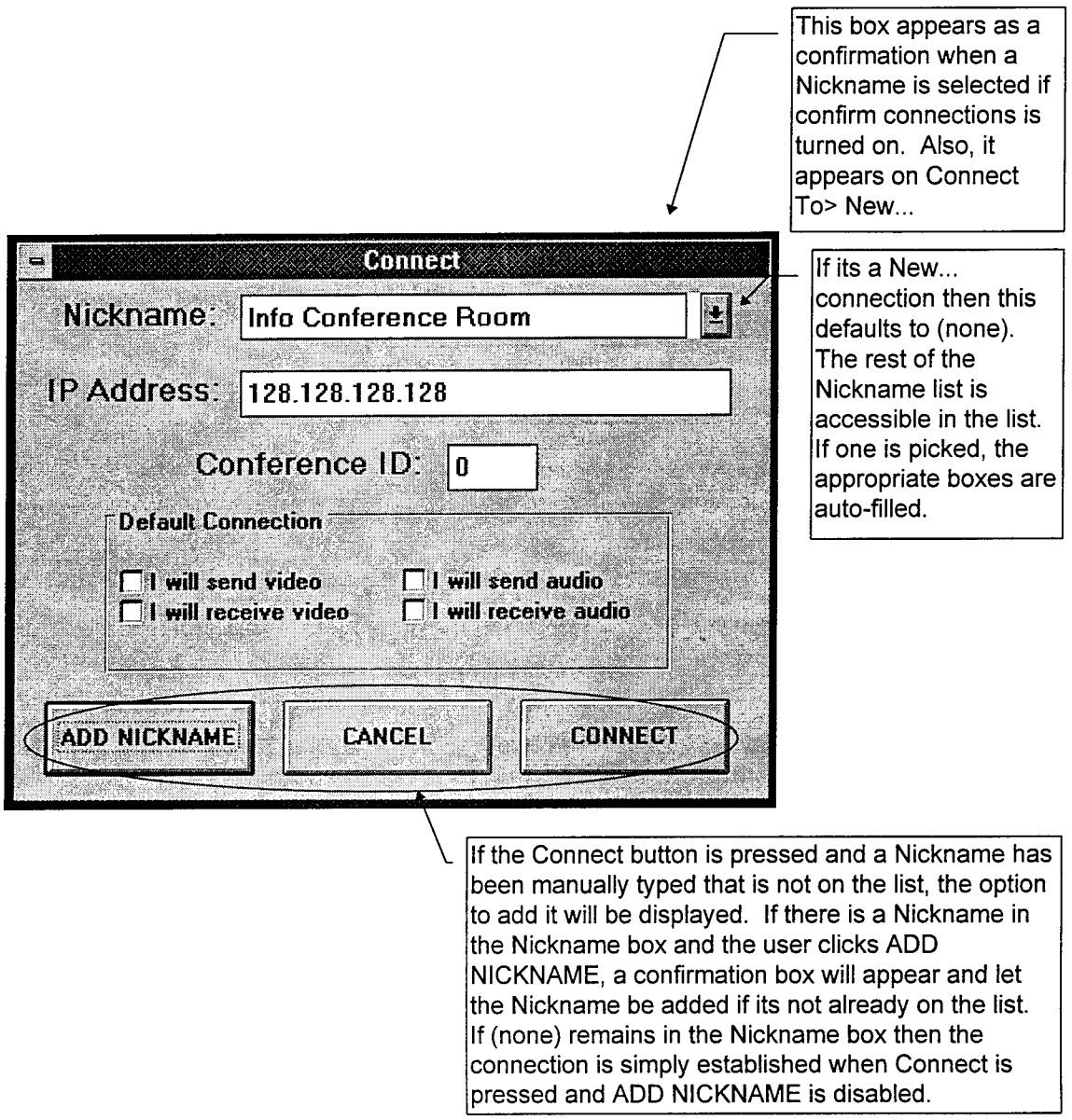


Figure 7: Redesigned Connection Dialog Box

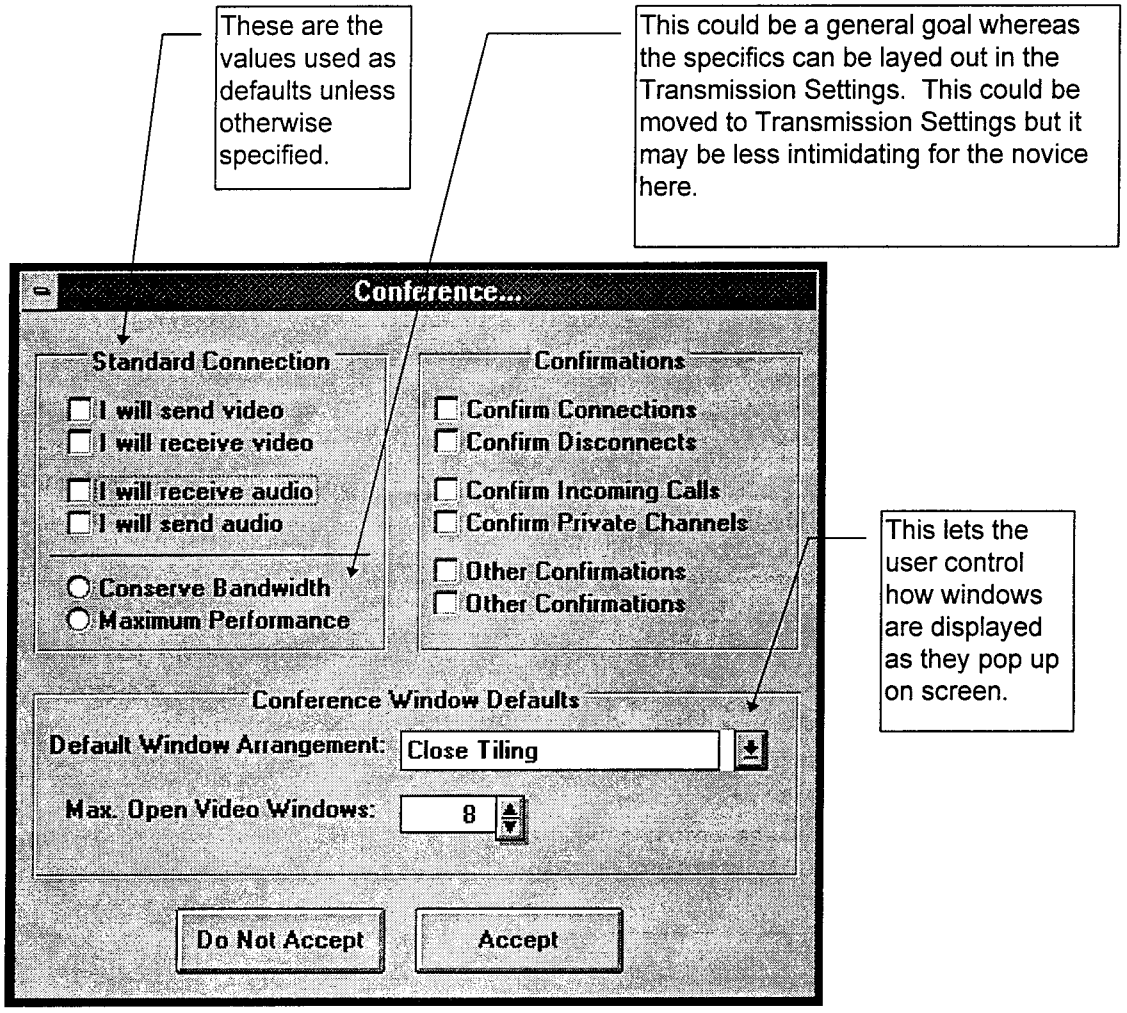


Figure 8: Redesigned Conference Settings Dialog

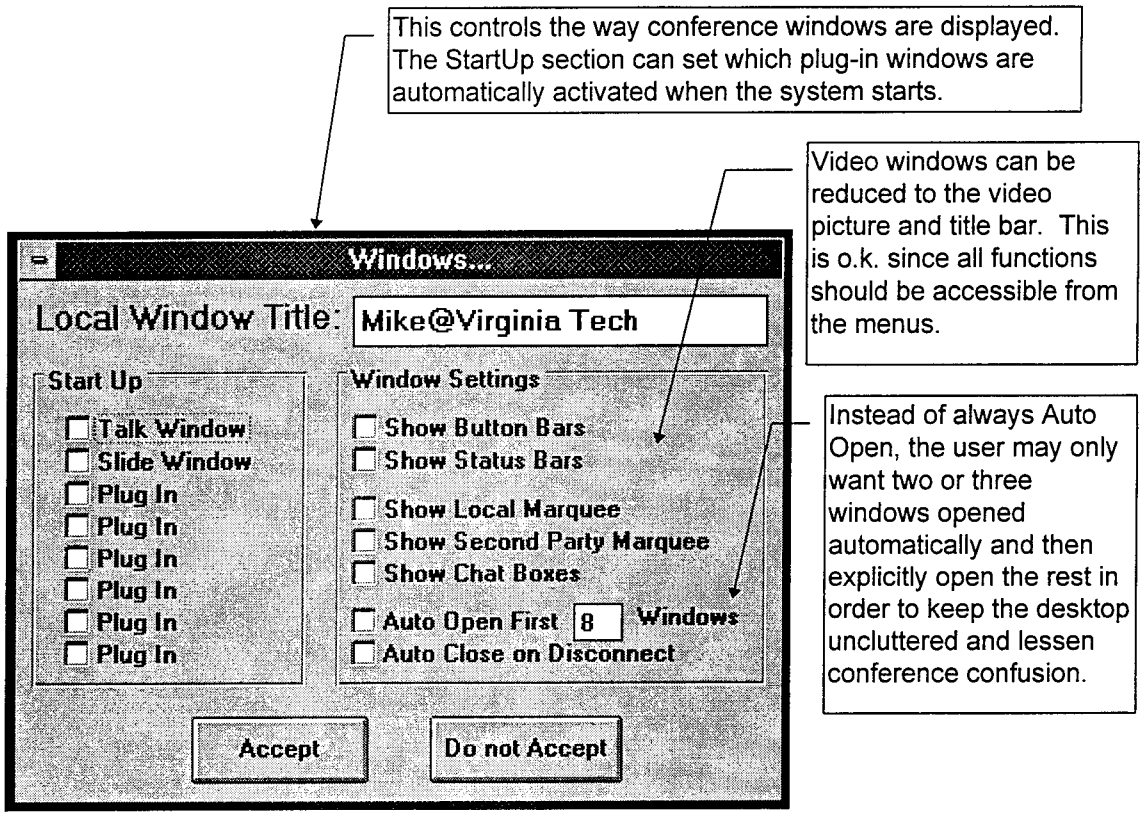


Figure 9: Redesigned Window Settings Dialog Box

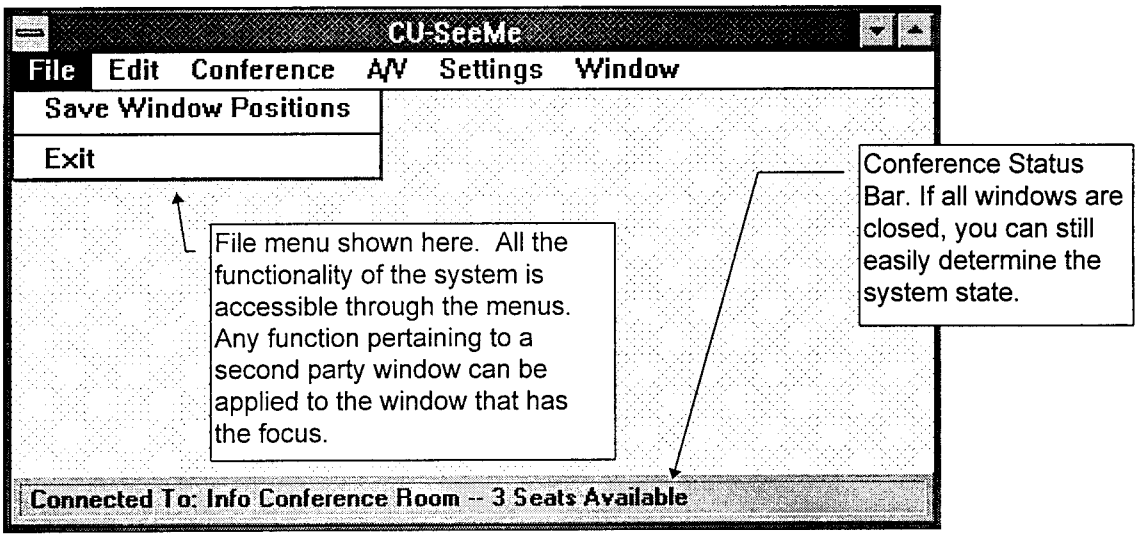


Figure 10: Redesigned Menu Structure (File)

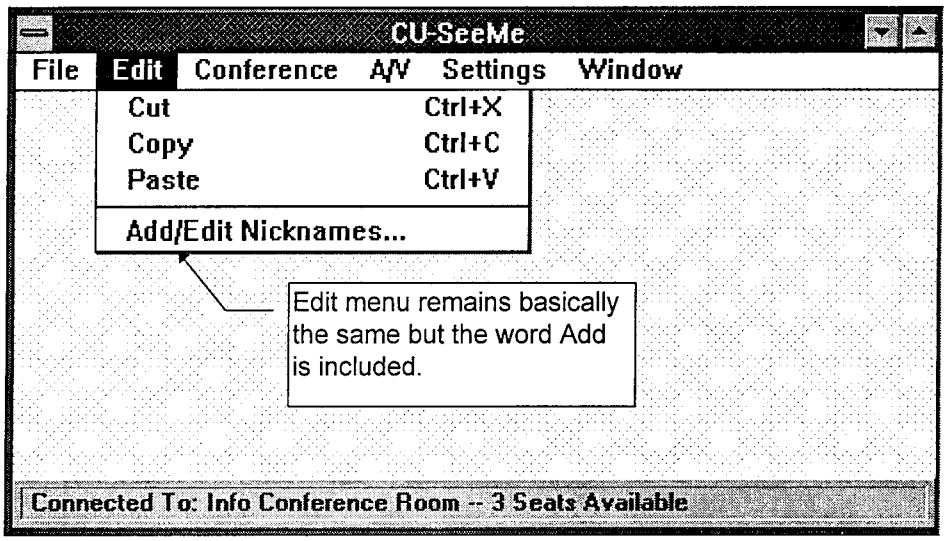


Figure 11: Redesigned Menu Structure (Edit)

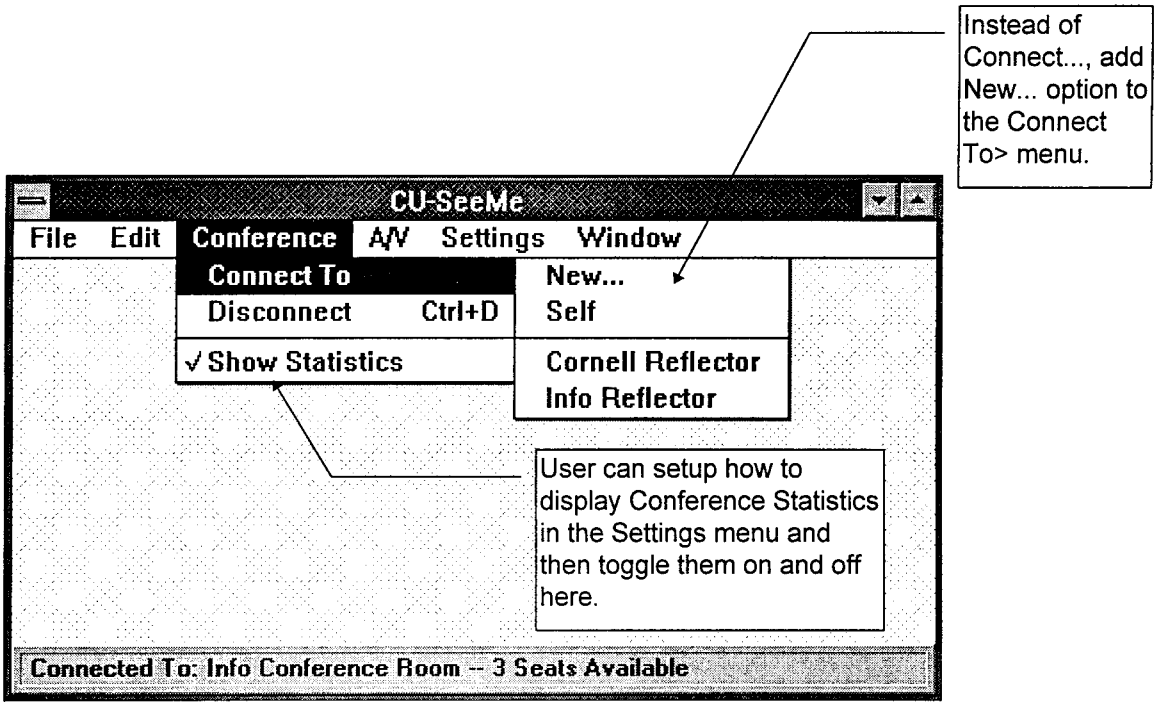


Figure 12: Redesigned Menu Structure (Conference)

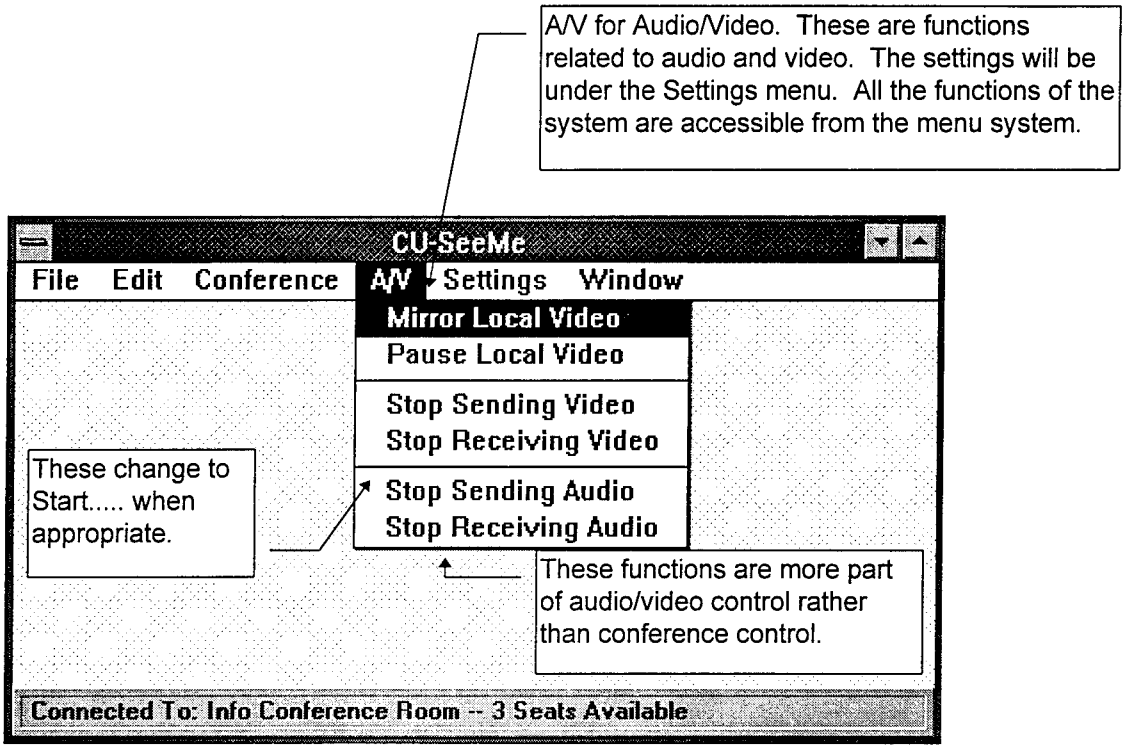


Figure 13: Redesigned Menu Structure (A/V)

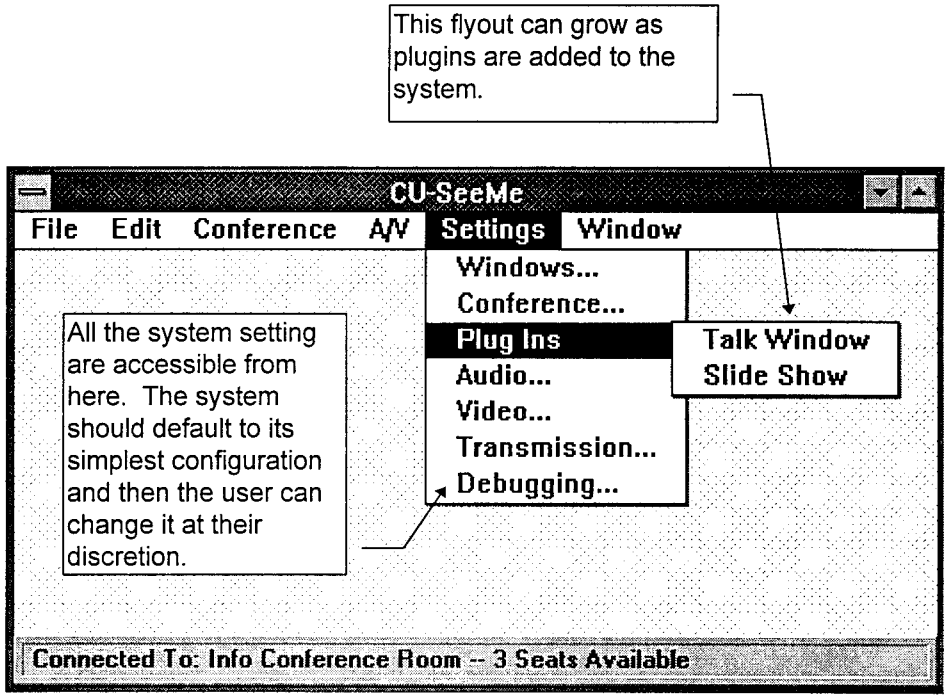


Figure 14: Redesigned Menu Structure (Settings)

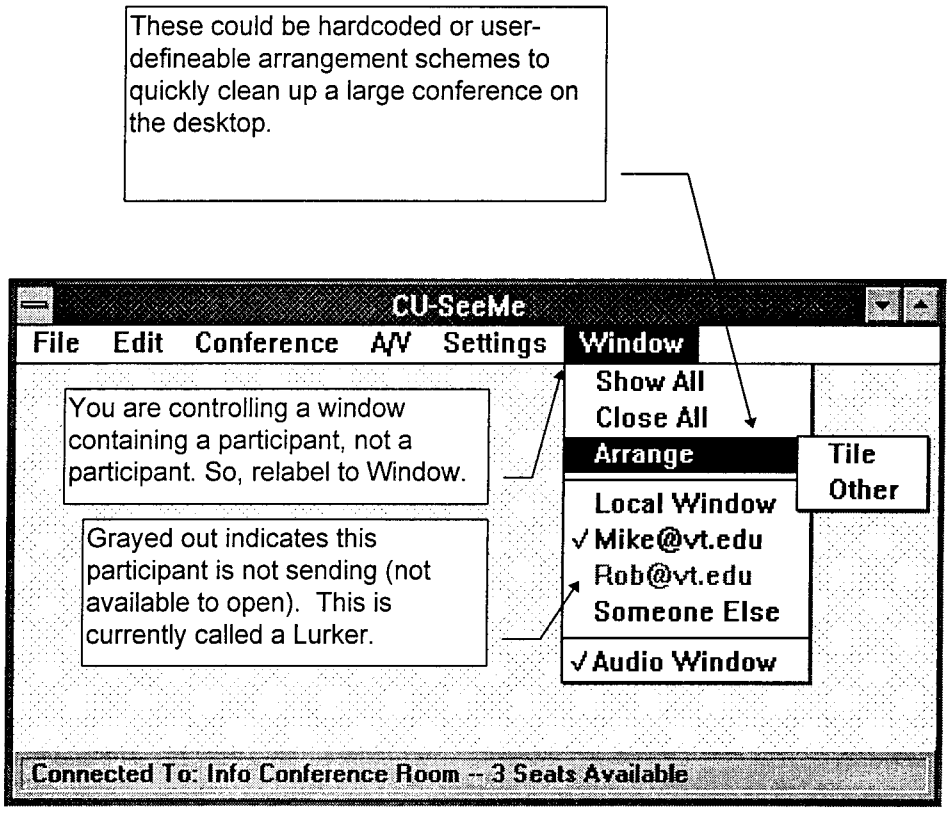
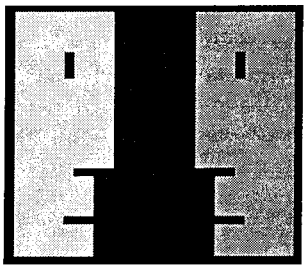


Figure 15: Redesigned Menu Structure (Window)

Addendum

This document was produced as part of the testing performed on CU-SeeMe.

CU-SeeMe



USERS MANUAL

Written By: Mike Bibeau

Virginia Polytechnic and State University

January, 1995

This manual has been written as of version .70b15. For further information on the most current version, refer to the README file supplied with the software.

(Thanks to Rob Mohn for help in constructing this document.)

TABLE OF CONTENTS

TABLE OF FIGURES

1. System Overview

Cu-SeeMe is a network based videoconferencing package that uses TCP/IP protocol. Currently, versions for the Macintosh platform, Power Mac, and PC are all available. The Macintosh and Power Mac versions are identical, but the PC version has taken its own development path and differs both in functionality and feel from the other two. Following is a complete description of the Macintosh/Power Mac version of CU-SeeMe. There does not exist any kind of user manual for this software as it is still in development. This section will help to clarify just where the software stands at this point so that we can better direct where the software should go in terms of its interface development. This manual has been written to accompany Formative Evaluation testing being conducted at Virginia Polytechnic and State University.

1.1 History[1]

CU-SeeMe originated at Cornell University. It was originally written for the Macintosh by Tim Dorcey, with help from Richard Cogger (Cornell University's Information Technology Dept., CIT), Scott Brim (Cornell University's Medical Colleges, CUMC), and John Lynn (CUMC). The project has received sponsorship from Richard Cogger and the CIT and began receiving funding from the National Science Foundation, NSF, on Oct. 1, 1993. The program has gone through several Beta versions with the current Macintosh version, 0.70b15 as of January, 1995.

1.2 Types of Connections

CU-SeeMe supports two types of conference connections.

[1] CU-SeeMe0.70b15 README file.

1.2.1 Point to Point

A point to point connection involves two parties who connect directly with each other. A point-to-point conference is conducted very much like a normal telephone call.

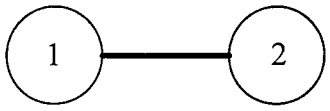


Figure 1: Point-to-Point connection diagram

Both of the two parties must have CU-SeeMe running and not connected. One party then "calls" the other party by using the other party's *IP address*, which is like the phone number of the computer. Once the connection is established, no one else can join the conference and neither party can connect to anyone else. Like a telephone call, once one of the two parties disconnects, both parties are disconnected and the conference session terminates. Since a party can elect to either not send or not receive signals, it is possible to set up a the conference like a lecture where one party, the lecturer, sends but doesn't receive, and the other party receives but doesn't send.

1.2.2 Multiple-Party

Multiple party conferences are conducted on CU-SeeMe by using a third computer known as a *Reflector*. This is very similar to using a telephone company switch to conduct a telephone conference call. Each party in the video conference makes a point-to-point connection with the same computer, in this case a reflector. The reflector simply receives all the signals and then transmits each signal to every other party that is connected to the reflector.

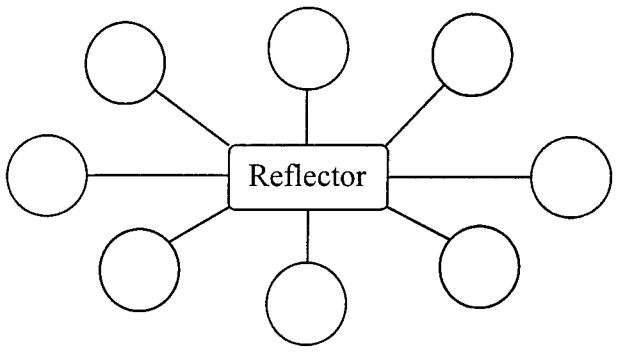


Figure 2: Multi-party connection diagram

Just as with a point-to-point conference, parties can elect to not send or to not receive. The NASA TV site, sends its signal to the reflector that transmits it to the public, but the site does not receive signals. Anyone connecting to watch NASA TV must connect in receive-only mode and the NASA reflector operates in send-only mode. There is only one party sending a signal, the NASA TV site, and any number of parties, the audience, receiving the signal.

Using a third computer to bear the load of a multiple party conference works well but the third computer must be a UNIX workstation running the reflector software. Since most people do not keep a UNIX workstation, the only way to popularize the use of CU-SeeMe conferences is through the use of public reflectors. Many universities and other organizations allow the general public to use their reflector sites when they are not being used otherwise. The following is a shortened list of reflector sites that can be used with CU-SeeMe for multiple-party conferencing. Note that in many cases use must be coordinated through the individual in charge of the reflector computer.

Reflector List	IP Address
Cornell Univ.	132.236.91.204
Cornell Univ.	192.35.82.96
GTE	132.197.10.74
Univ Ulster	193.63.68.162
NASA Select USA	139.88.27.43
NASA Select Europe	158.36.33.5
NYSerNet	192.77.173.2

Ostfold College Two	159.36.33.5
Penn State	128.118.3.57
Univ. Indiana State	139.102.70.201
Univ. Kansas	129.237.247.160
Univ. Michigan	141.214.20.107
Univ. Murdoch	134.115.224.60
Univ. N. Carolina	152.1.57.56
Univ. Ohio State	128.146.116.8
Univ. Penn	130.91.72.36
Univ. Sao Paulo	143.107.225.6
Univ. Singapore	137.132.9.61
Univ. Texas	128.83.108.14
Weizmann Institute	132.76.64.143

As of January 95, a current list of Reflector sites can be found on the World Wide Web at **http://gated.cornell.edu/pub/video/CU-SeeMe_Nicknames** or at **http://pixel.cs.vt.edu/mike/reflist.html**.

1.3 System Performance

System performance is a very important issue in making videoconferencing via computers and the Internet a useful technology. CU-SeeMe provides relatively good performance over a direct network connection on a point-to-point conference. Beyond that, the performance is useable, but leaves much room for improvement. As researchers find better and faster ways to compress and send the large amounts of data needed for a conference, system performance will increase. Also, better hardware is constantly evolving which also contributes greatly to overall system performance.

1.3.1 Frame Rate

Frame Rate refers to the speed at which the moving video picture is updated. The frame rate on CU-SeeMe stays fairly high when there is no connection established, however, once a conference is started frame rates are usually in the range of 3-7 frames per second. Generally, on a multiple party conference, each participant will see 1-5 frames per second. A video picture that is

constantly changing, such as NASA TV will generally stay very low, like 0-2 frames per second. Louis Lumière's slightly flickering cinématographe of 1895 ran at 16 frames per second, and a 70mm film of today runs at 24 frames per second.[2]

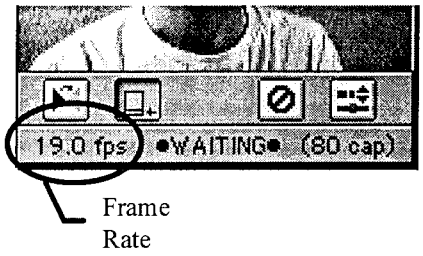


Figure 3: Frame Rate 19.0 fps (unconnected)

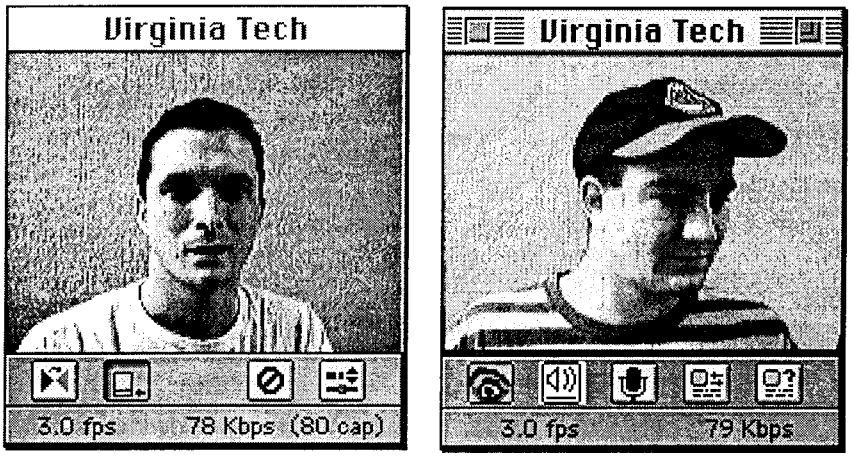


Figure 4: Frame Rate 3.0 fps (connected)

1.3.2 Picture Quality

CU-SeeMe supports 16 gray colors at a resolution of 160x120 or 320x240. Even with good frame rates, the video quality can sometimes degrade. The reason for this is that, in order to save

[2] Encyclopedia Brittanica, Vol 24. Encyclopedia Brittanica Inc, Chicago. 1992.

network resources, CU-SeeMe uses an algorithm that only updates parts of the screen that change. It breaks the screen into 8x8 boxes of pixels, so the video window consists of a set of 300 boxes with each box containing 64 pixels. If the screen is updated but one part, or box of pixels, gets lost in the transmission, the video picture degrades. The box that was lost will be resent after either a preset time frame or when it changes again. Even with a good frame rate, lost data will cause annoying video artifacts during the time between when the data was lost and resent. This happens more with video windows that contain a large amount of activity.

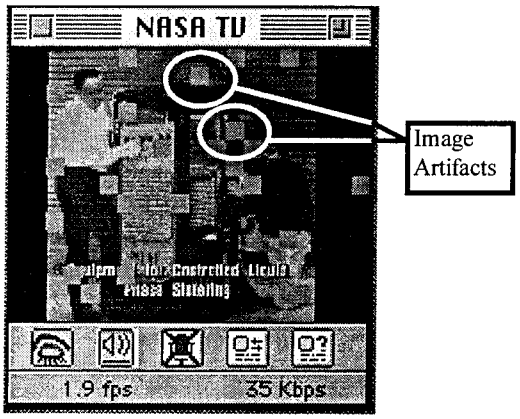


Figure 5: Image Artifacts

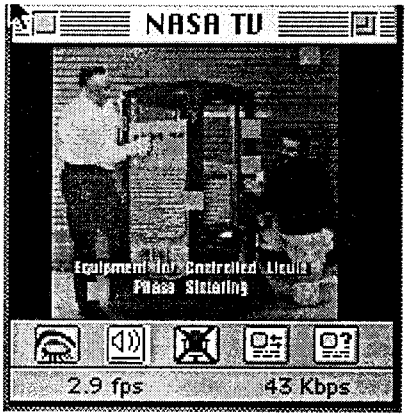


Figure 6: Artifacts Healing

1.3.3 Audio Quality

Once configured properly, audio performs quite well on CU-SeeMe. When data is lost in an audio transmission, however, the results can be unintelligible at the other end of the transmission. Unlike the video picture which will show an image artifact when a piece of video is lost, the lost audio cannot be recovered since each individual piece is time dependent on the previous piece. Since the audio transmission must be linear and ordered, lost data will manifest as broken sentences to the receiver that cannot be repaired or resent.

2. CU-SeeMe Hardware Requirements

The following hardware requirements list comes from the CU-SeeMe documentation: [3]

2.1 To Receive video

- Macintosh platform with a 68020 processor or higher
- System 7 or higher operating system (it may run on system 6.0.7 and above)
- Ability to display 16 level grayscale (e.g. any color Mac)
- an IP network connection
- MacTCP
- Current CU-SeeMe application software

2.2 To Send video

- The specifications to receive video mentioned above
- Quicktime installed
- A video digitizer (with vdig software) and a camera;
- **Supported as of 0.7b15**

ONE OF

Video Spigot hardware (street price approx. $380)

AV-Mac (vdig built into system)

ComputerEyes/RT SCSI port digitizer

PLUS

Camera with NTSC 1vpp output (like a camcorder) and RCA cable

[3] CU-SeeMe0.70b15 README file.

OR

Connectix QuickCam serial port camera

2.3 To Receive Audio

You need an audio-capable Mac to receive audio signals.

2.4 To Send Audio

You must have at least Sound Manager 3.0 installed in the Mac System Folder in order to send audio. You also need a microphone that is configured as the Sound In source in the Sound Control Panel.

3. Setting Up CU-SeeMe

3.1 Obtaining the Software

CU-SeeMe can be obtained direct from the source via anonymous FTP at cu-seeme.cornell.edu.

Be sure to download the file as a binary or use the "automatic" setting on your FTP software.

FTP site:	cu-seeme.cornell.edu
UserID:	anonymous
Password:	<none>
Directory:	/pub/video

Be sure to read the README files. They will tell you what to download since it changes as the software is updated. MacTCP can also be obtained at this site but be sure to read the license agreement since it is commercial software.

3.2 Installing the Software

Downloaded as a binary, the software is ready to run. Put the files in an appropriate folder on your Macintosh desktop and you are ready to run the program.

3.3 Machine Configuration (Control Panels, etc...)

3.3.1 Audio

You must have *SoundManager 3.0* extension installed for the audio to function properly. This is NOT needed on the PowerMac version. Be sure the microphone is connected and set the volume to the desired level in the *Sound Control Panel* on the Mac. Control Panels are found in the System Folder. Problems can be caused with audio if the Sound Control Panel is not set properly. In the Sound Control Panel, be sure that *Sound In* is set for the external microphone, and you may need to lower the sampling rate in *Sound Out* if you receive errors when trying to use the audio.

3.3.2 Video

If you do not have an AV-Mac, you must have a *VideoSpigot* installed with the *Quicktime* extension and the *Spigot VDIG* component. The screen must be switched to a resolution that includes 16 gray levels. This can be set in the *Monitors Control Panel* on the Mac. The best performance will be obtained with the lowest color depth, i.e. settings the Monitor Control Panel to 16 grays. However, a standard setting of 256 colors works well on most systems. An error that reads "I found a digitizer component, but am unable to digitize. Continue in receive-only mode?" can usually be fixed by lowering the number of colors to 256 or less in the Monitor Control Panel.

4. System Tour

4.1 Starting the CU-SeeMe software the first time.

When the software is FTP'ed as a binary, it is in executable format. Simply launch the program by double-clicking on its icon wherever it was installed. If it is the first time running the program, a *Preferences Box* will be displayed.

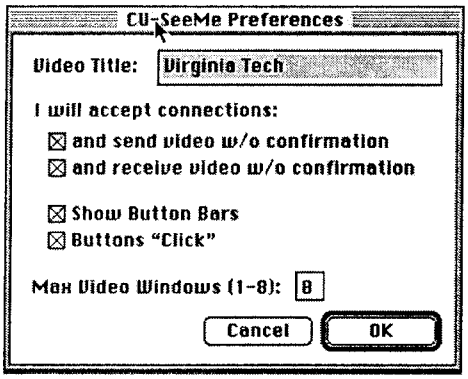


Figure 7: CU-SeeMe Preferences

Enter the name you wish to appear at the top of your video window in the Video Title box. This name will appear on your window on other computers during a conference. For now, you can leave the rest of the settings at their defaults. After launching the program and filling in the Preferences Box (if its the first time running), you will see an Audio Window on the screen if the audio is installed properly. If you do not have Sound Manager 3.0 installed, you will get a message indicating that audio will work in receive only.

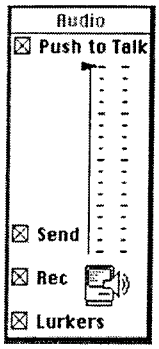


Figure 8: CU-SeeMe Audio Window

You will immediately know if you have your video hardware set correctly when you start CU-SeeMe because on startup (after filling in the Preferences box on first startup) you will see one of the following three situations.

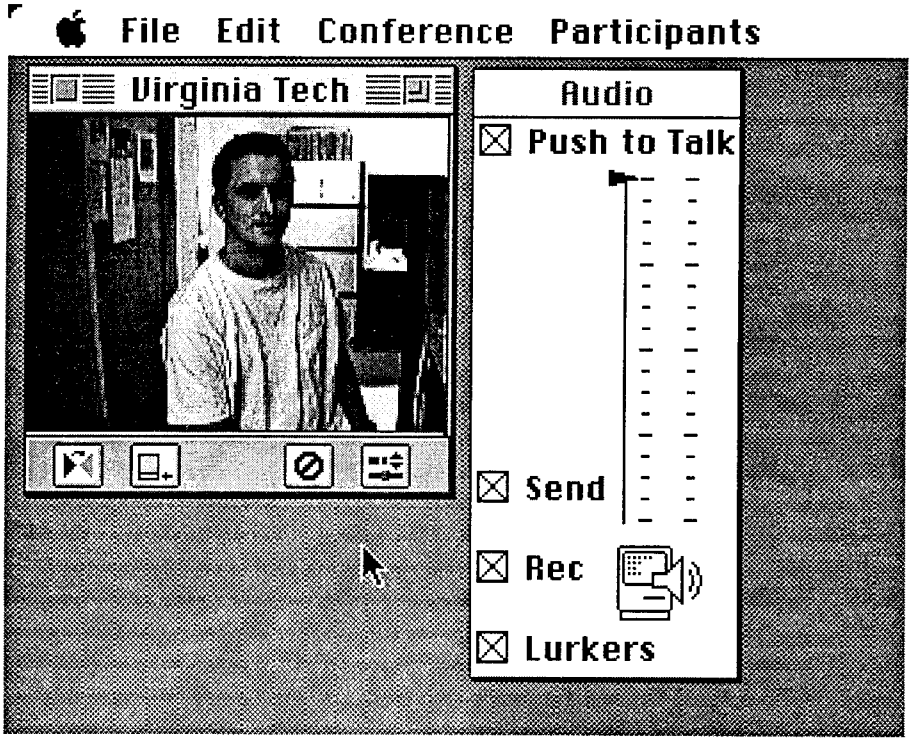


Figure 9: CU-SeeMe Startup Screen

4.1.1 Fully Working

If the screen contains a video window in the upper left corner that is displaying your picture, like the one above in Fig. 9, and you have an audio window like Fig. 9, then you have everything installed properly. That means you have either and AV-Mac with the camera plugged in or you have VideoSpigot installed with Quicktime, the SpigotVDIG component and a camera, and you have the proper audio components.

4.1.2 Send/Receive working but no camera.

If the screen looks similar to Fig. 9 but the video is solid black, then you have everything installed properly, but you have no camera plugged in and working.

4.1.3 Receive-only

If the program menu appears but no window appears in the upper left corner, then the program could not find the proper video components needed to send video. You can still run the program in Receive-only mode and sit in on a conference but you cannot participate. Once you connect to a site or another party, you will seen their window displayed on your screen but they will not see a window from you. However, you may be able to send\receive audio-only if the audio is installed properly.

4.1.4 Testing the Software

Once the program is running on your computer, you need to test that it connects properly. The first way to test the program is to connect to yourself. Select the Conference Menu from the menu bar and move the mouse down to Connect To. A fly-out menu will appear with the word Self at the top. (You can add names to this menu later) Select Self from the fly-out menu. A second video window should now appear that has an identical picture and title to your local video window. If you cannot establish a connection to Self, then check that your network is functioning properly.

Choose Conference-Disconnect from the menu bar and close the second window by clicking in the small box in the upper left corner of that window.

The true test of the software is to connect to another computer and conduct a conference. You must have and IP address for another computer running CU-SeeMe, or the IP address for a CU-SeeMe reflector. To test CU-SeeMe, it is better to start out by connecting directly to another computer instead of a reflector, since you will not know ahead of time if someone else is connected to the reflector. If you do not have another computer available running CU-SeeMe then you can try some of the reflector sites from the list in section 1. Eventually, you will find a site with at least one person connected that you can talk with. NASA TV is a good place to test that you can receive both video and audio since it runs continuously, but you will not be able to talk to anyone.

4.2 Conference Windows

4.2.1 Window Update Speed

Video Windows in CU-SeeMe are normally updated by writing directly to the screen using optimized algorithms. However, if part of a window is covered, or the window is mirrored, the Macintosh Quick Draw algorithm is used, which is much slower. If a window is being updated with the slower Quick Draw routine, a black border will appear around the window. You will get the best window performance by not using the mirror function, making sure the entire window is on the screen and uncovered, using standard (160x120) resolution, and setting the screen color depth to 16 grays.

4.2.2 Local Window

The Local Window is what is referred to as the window displaying your video on your screen.

Button Bar

There are four buttons on the local window. The button bar can be toggled on and off in the Preferences window under the Edit Menu.

 **Mirror Button**

The Mirror Button flips the video in the Local Window so that it looks the same as if you were looking in a mirror. This makes it more natural to position yourself in front of the camera but unfortunately it also flips your text line. So, now all the text you type in your local window will appear backwards to you. **The reversal only affects your local window. All the other conference participants will still see a normal picture.**

 **Local Info Button**

The info button toggles the information line below the row of buttons.

 **Pause Button**

This button pauses your stream of video. You will remain connected, but your image will appear still to all other parties.

 **Local Settings Button**

This button will toggle the display of the settings box beneath your video window.

Local Information Line

The Information Line appears directly below the button bar. It displays three pieces of information about the local window.

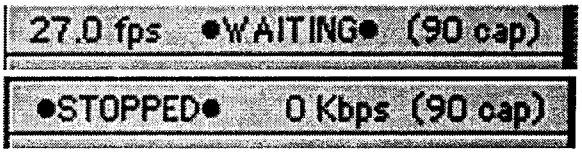


Figure 10: Two Info Line Examples

1. The leftmost statistic is the framerate. It will indicate STOPPED when no frames are being sent. When you are not connected, it will indicate the frame-grabbing rate of the local window.
2. The middle statistic indicates the bandwidth being used, in kilobits per second (kbps). This will indicate WAITING when you are ready to start a conference, and will indicate 0 kbps when you are not sending.
3. The right statistic indicates the cap on your bandwidth use. The number indicated shows the maximum kbps that you will send. This value can be changed in the Transmission Settings in the Software Settings Box.

Software Settings Box

By pressing the far right button on the button bar in the local window, you will toggle the Software Settings Box display. There are currently five settings modes which are selected via the dropdown list at the top of the Settings Box.

Local Picture Settings (Brightness/Contrast)

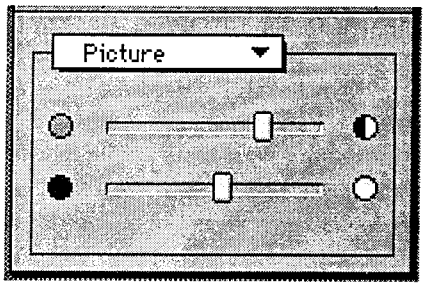


Figure 11: Local Picture Settings

You can adjust the brightness and contrast of your local picture with the Picture Settings. The top slider adjusts contrast and the bottom slider adjusts the brightness.

Audio Transmission Mode

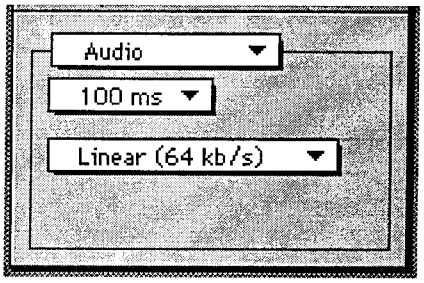


Figure 12: Audio Settings

This box allows you to choose the compression scheme used for audio compression. The numbers in parenthesis indicate the bandwidth used by the scheme. The performance of each scheme will depend somewhat on the speed of your system, but generally one that uses less bandwidth, i.e. Δ-mod(16kb/s), is preferable.

Transmission Parameters

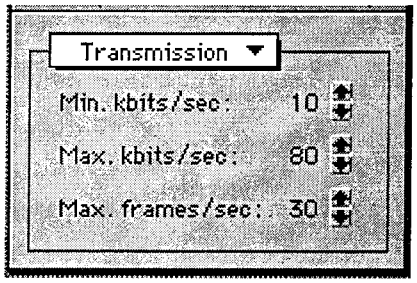


Figure 13: Transmission Parameters

This box allows you to control your bandwidth use. You can set the minimum and maximum bandwidth that CU-SeeMe will use by modifying the Min. Kbits/sec and Max. Kbits/sec settings. You can also limit the frame rate which will effectively lower your bandwidth use. CU-SeeMe will automatically lower your Max. Kbits/sec when it receives reports of packet losses, indicating that there is heavy network congestion. It does this to ease the load on the net when traffic is heavy. When the loss reports stop, CU-SeeMe will raise the Max. Kbits/sec back up to the preset value. The rate used by CU-SeeMe will *always* remain within the parameters you specify. Currently, you have no control over the bandwidth use of incoming signals other than closing their window, at which point you will receive **no** packets from that individual.

Compression and Resolution Settings

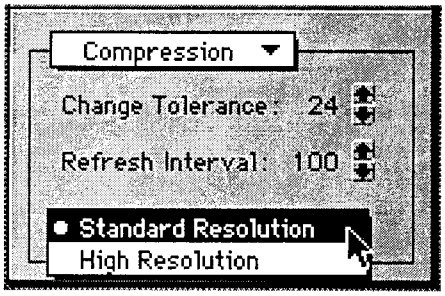


Figure 14: Compression and Resolution Settings

This box allows you to set the change tolerance for a portion of the window before it is resent. A setting of zero will re-send on the slightest change. The refresh interval is the time interval to automatically send an unchanged box of pixels to heal image artifacts. A very low change tolerance is going to result in poor transmission performance since the entire image will be constantly updated, and a too high setting will result in poor picture quality.

Video Input Settings

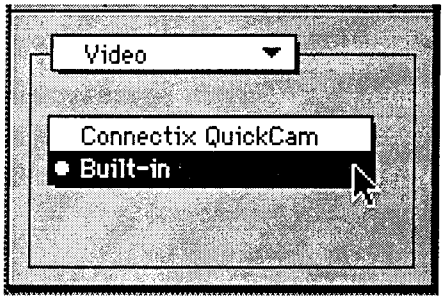


Figure 15: Video Input Settings

You can choose to use either the built in video camera (standard camera) or a Connectix Quickcam serial port frame grabber/camera. The Connectix option will not be available if the system has not detected the proper drivers installed on your system. (The Connectix software must be installed with the OEM disks)

4.2.3 Second Party Windows

Button Bar

 **Visible Window Indicator - Eye**

This is actually not a button at all. The "eye-con" indicates whether or not the person in a second party window can see you on their screen. The closed eye indicates that you are not visible on the screen of the person in that window and the open eye indicates that you are visible to them.

 **Audio Enable/Disable and Transmit Indicator - Speaker Button**

Pressing the speaker button down will disable the audio for a specific second party window. You will not be able to hear the person in that window, but will still hear the other conference participants. When you disable the audio for a specific party, they will see an X over the microphone icon on your window that is displayed on their screen indicating to them that they cannot talk to you. When you are receiving audio from a specific party in a conference, the speaker icon on their window will turn gray.

 **Private Channel Selector and Receive Indicator - Microphone Button**

Pressing the microphone button on a second party window will establish a private talk channel to the person in that window. Your audio will only be transmitted to the person in the window whose microphone button you pressed. They will still hear all other parties in the conference as well as you. An X displayed over the microphone button on a second party window indicates that the person in that window cannot hear you. They either do not have audio capability or have pressed the speaker button on your window that is displayed on their screen.

 **Transmission Statistics Button**

The statistics button will bring up a box beneath a given parties window that displays statistics about the communication between you and that party. See Statistics Box below.

 **Second Party Information Button**

The information button on a second party window will display an information box below the window indicating relevant information about the party in that window. Currently, name and IP address are displayed, but the CU-SeeMe developers have hinted at much more to come in the future.

Second Party Information Line

The information line on a second party window displays the same kind of information as the information line on the local window. The line is displayed directly beneath the video picture of a second party window and cannot be removed like the information line on the local window. The left side of the line displays the frame rate of the window, the right side shows the current bandwidth being used, and the line will display DISCONNECTED if that window is disconnected from your computer.

Statistics Box

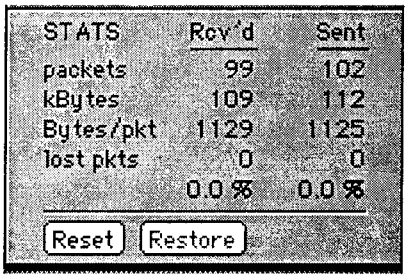


Figure 16: Statistics Box

The statistics box for a second party window gives detailed information about the conference transmission between you and the person in that window. The information here relates to network performance during the conference.

Info Box

The information box on a second party window displays the IP address of the computer generating that window and the version number of the CU-SeeMe software running on that computer.

4.2.4 Audio Window

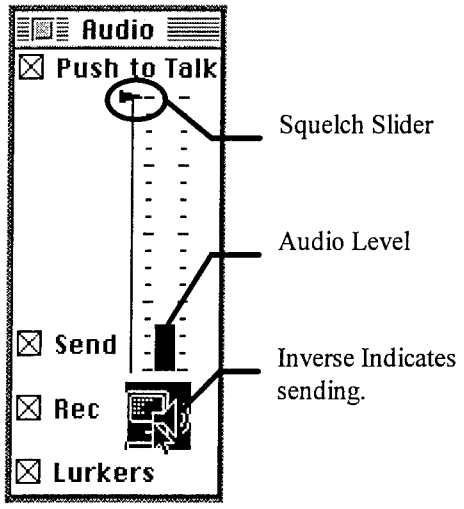


Figure 17: Audio Window Components

Check Boxes

Push to Talk

This box controls whether you will send audio continuously or in Push-to-Talk mode, like on a CB radio. An x indicates that to talk you must click and hold the mouse button while the cursor is in the audio window.

Send

This toggles whether or not you send out an audio stream. An x indicates that audio will be sent.

Rec

This toggles whether or not you receive audio from the other parties. An x indicates you will receive audio.

Lurkers

A Lurker is a party using a stand alone Maven client (an audio-only program) or using CU-SeeMe and transmitting audio-only. An X in this box indicates that you will accept audio from these "Lurkers".

Squelch Slider

The vertical squelch slider allows you to set the minimum level of audio to actually be transmitted. When you speak into the microphone, an audio level indicator will move up and down, indicating the audio level. Only when the audio level indicator is above the squelch slider will the audio be transmitted. When using push-to-talk, you should turn the squelch down low, but in continuos audio mode it is useful to slide the squelch up. Setting the squelch so that your normal speaking voice produces a level just above it, will cause only your talking to be transmitted and not any background noise in the room.

Speaker Icon

The speaker icon indicates when audio is being transmitted. When audio is actually being sent, the icon will inverse. If the squelch is too high or you are in push-to-talk mode and not pushing the mouse button in the audio window, your audio will not be sent. That will be indicated by the speaker icon not changing to inverse.

4.3 Menu Bar

4.3.1 File

Open

This option currently is not used for anything.

Save Window Positions

Windows are opened in some sort of order when a conference is started. Saving window positions will remember the locations of each window relative to the order they were opened. Who is in a given window is irrelevant. The first second party window opened will open in the same position the next time a conference is started, etc.

Close Window

Closes the currently active window.

Save

This option is currently disabled.

Quit

Quits CU-SeeMe

4.3.2 Edit

The Edit menu contains standard Macintosh Edit commands like Cut, Copy, Paste. These can, for example, be used to copy IP addresses from a typed list into CU-SeeMe to begin a conference.

Edit Nickname

Edit Nickname allows the user to add or modify a name on the Nickname list. Using a nickname list eliminates the need to enter conference information for a connection that is used frequently. Each connection on the list must have a name and IP address. The two check boxes, *I will send video* and *I will receive video* relate to how the conference will start when a connection is made. For example, you may want to turn off (uncheck) "I will send video" for a certain site so that when you connect to that site you can see who is there before making your presence known by selecting Start Sending from the Conference Menu.

Preferences...

The Preferences box is where you set the title that will be displayed above your conference window. The first two check boxes relate to what will happen when someone else establishes a connection with you. There is no option to not accept connections, aside from turning off CU-SeeMe. Unchecking one or both of the first two check boxes will cause a dialog box to appear on your screen when someone makes a connection with you. The dialog box will ask you whether you wish to begin sending or receiving video.

The bottom two check boxes relate to the button bars on the conference windows. The *Max Video Windows* box is where you can set the maximum number of windows allowable to open on your screen. In a multiple party conference, it is possible for there to be several people sending a signal, so this will limit how many will pop up on your screen when you connect. The more windows you have open, the slower the system performance will become.

4.3.3 Conference

Connect...

Connect... is used to manually connect to a conference. The *Conference ID* is used when an exclusive conference is being held on a reflector site. The individual controlling the reflector can cause the reflector to reject all connections that do not have the correct conference ID. Normally, it is set to zero and most reflector sites do not require a conference ID.

Connect To

Connect To> allows you start a conference by selecting from the Nickname list. When a name is selected, a dialog box will appear in case you wish to change any settings before connecting.

Disconnect

This disconnects you from whatever conference you are currently connected.

Stop Sending

This will halt your video but keep you connected to the conference in receive only mode.

Stop Receiving

This will halt all video from other parties but you will still be sending your video.

Audio Window

This displays the Audio Window. It is the only way to call up the Audio Window if it has been closed.

4.3.4 Participants

The *Participants* menu is used to open conference windows that have been closed. CU-SeeMe can only have eight windows open at any one time, however, a conference can have more than eight participants. The Participants menu will show up to 16 participants.

Show All

This opens a window for all participants, up to eight, that are sending video.

Close All

This closes all the conference windows but does not disconnect. Remember, simply closing a window does not disconnect a conference.

Local Video

This is to access your local window. Select it to reopen your local video window.

Visible Participants

Under the Local Video option will be a list of participants who are sending video. This is indicated by the empty box to the left of their name. Select one to reopen their video window on your screen.

Invisible Participants (Lurkers)

At the bottom of the list are participants who are connected to a conference but not sending (receive-only). You cannot open a window for these participants but will be able to see the names of who is "spying" on you.

5. How To

5.1 How To Connect to a Site and Start a Conference

5.1.1 Connecting to a New site

Choose *Conference* from the menu bar and select *Connect . . .* Enter the IP address of the site you wish to connect with in the IP address: box. Put an X in the *I will send video* and/or *I will receive video* boxes if you wish to both send and receive. Click on *Connect.*

5.1.2 Picking the site from the Nickname list

Choose *Conference* from the menu bar and select *Connect To.* Move the mouse to the flyout list at the right and choose the name of the site you wish to connect with or choose *Self.* When the dialog box appears, put an X in the *I will send video* and/or *I will receive video* boxes if you wish to both send and receive. Click on *Connect.*

5.2 How To Disconnect from a Conference

To disconnect from a conference, choose *Conference* from the Menu bar and drag the mouse cursor down to *Disconnect.* You will have to close each window that is on the screen manually by clicking the box in the upper left corner since they do not automatically disappear when they are disconnected from a conference.

5.3 How To Control Sending and Receiving Video

Without disconnecting from a conference, you can control if you will send your video stream to others or if you will receive video from others.

5.3.1 Starting or Stopping Sending Video

Select *Conference* from the menu bar. If you are already sending your video and wish to stop, select *Stop Sending*. If you are not sending and wish to start so that others can see you, select *Start Sending*.

5.3.2 Starting or Stopping Receiving Video

Select *Conference* from the menu bar. If you are already receiving video and wish to stop, select *Stop Receiving*. If you are not receiving and wish to start so that you can see others, select *Start Receiving*.

5.3.3 Pausing Video Without Disconnecting

Click on the  button in the local video window. Your video picture will be paused but will continue to send a signal so that you show up to others on the conference paused.

5.4 How To Use Audio

5.4.1 Transmitting Audio

If the *Push to Talk* check box has an X in it, then put the mouse cursor **anywhere** in the Audio Window except over one of the check boxes, press and hold the left mouse button, and begin speaking. The icon at the bottom of the Audio Window will turn inverted, indicating that you are transmitting audio, and you will see an audio level indicator moving up and down.

If the *Push to Talk* check box does **not** have an X in it, simply talk and you will see the audio level indicator moving up and down. Slide the horizontal triangle next to the audio level indicator to the desired level. The audio will not be transmitted until the audio level rises above the triangular slider. When the audio level is above the slider, the icon at the bottom of the audio window will invert, indicating that you are transmitting.

5.4.2 Setting a Private Audio Channel

Click on the  button in a second party window to establish a private channel with that party. The button will depress and get a red dot in the middle, indicating that **only** the party in that window can hear you. You will see X's over the microphones on all other windows.

5.4.3 Disabling/Enabling a Second Party's Audio

Click on the  button in a second party window to disable their audio. You will no longer be able to hear that party, but will still hear everyone else. The two small sound waves on the speaker button will disappear, indicating you cannot hear that party. Click on the button again to enable that party's audio and the sound waves will reappear on the icon.

5.4.4 Who is Speaking in a Multi-Party Conference

When you are receiving audio from a second party, the  button on their window will turn gray, indicating that they are speaking.

5.5 How To Use Nicknames to Maintain Connection Sites

CU-SeeMe has the ability to maintain a list of commonly used places to connect called a *Nickname* list. This allows you to enter the IP address one time, attach a name to it, and then you can select it from a list thereafter.

5.5.1 Adding/Editing Names on the Nickname List

To add names to the Nickname list, select *Edit* from the Menu bar and move the cursor down to *Edit Nickname*. A fly-out menu appears showing the current list. You can select a name on the list to edit or select *New* to add a new name to the list. Enter the Nickname and IP address. Put an X in the *I will send video* and/or *I will receive video* boxes if you wish to both send and receive. Click on *OK*.

5.5.2 Deleting Names from the Nickname List

To delete a name from the Nickname List, select *Edit* from the Menu bar and move the cursor down to *Edit Nickname.* Select the name off the flyout list that you wish to delete. When the dialog box appears, click on *Delete.*

5.6 How To Change System Settings

Click on the  button on the local video window to toggle the settings box. Click on the label indicating which settings box is open and you will see a dropdown list like this:

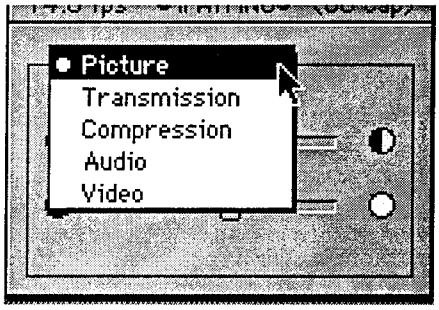


Figure 18: Settings Menu

Choose an item from this dropdown list to bring up the desired settings box.

5.6.1 Setting Video Brightness and Contrast

Bring up the Picture settings box. Slide the top controller left/right to adjust contrast and slide the bottom controller left/right to adjust brightness.

5.6.2 Setting Transmission Parameters

Bring up the Transmission settings box. Click on the up/down arrows next to each parameter to adjust to the desired level.

5.6.3 Setting Resolution and Compression

Bring up the Compression settings box. Use the arrows next to the parameter to adjust compression settings for Change Tolerance and Refresh Interval. Click in the Resolution box at the bottom to open a pull down list and select either Standard Resolution or High Resolution.

5.6.4 Setting Video Camera Input

Bring up the video settings box. Click in the pull down menu box and select either Connectix QuickCam if you are using the Connectix or Built-In if you are using a regular video camera.

5.6.5 Setting Audio Compression

Bring up the Audio settings box. Click in one of the two menu boxes to change the settings. Choose the sample spacing from the upper pull down menu and the compression scheme from the lower pull down menu.

5.7 How To View Conference Information

5.7.1 Toggling Local Video Information

Click on the  button on the local video window to toggle the information line on and off.

5.7.2 Viewing Transmission Statistics

Click on the  button on a second party window to toggle transmission statistics for that party on and off.

5.7.3 Viewing Information on a Second Party

Click on the  button on a second party window to toggle the information box for that party on and off.

5.8 How To Manage Windows

5.8.1 Changing the Name Above the Local Video Window

Select *Edit* from the menu bar. Choose *Preferences...* from the menu and in the *Video Title* box, enter the name you wish to appear above your video window. This name will appear to other participants. If you change the name while connected to a conference, you must disconnect and connect again for the change to be visible to other parties.

5.8.2 Opening the Audio Window

Select *Conference* from the menu bar. Choose *Audio Window* from the menu to open the Audio Window if it has been closed.

5.8.3 Sizing a Video Window

All the conference windows have non-sizable borders but you can click in the upper right corner of a window to make it larger. Clicking in the upper right corner does not increase the picture resolution, it only increases the size of the picture by guessing the extra pixels using linear interpolation.

5.8.4 Moving a Window

All the windows can be moved just like any Macintosh window. Click and hold in the title bar and then drag the window outline to the desired new location.

5.8.5 Closing a Window

Windows can be closed just like any Macintosh window. Click in the box in the upper left corner of a window to close the window. The individual at the other end of the conference whose window is closed on your screen will know by the appearance of a closed-eye icon that will appear in your window on their screen. Simply closing a window does not disconnect that individual from the conference.

5.8.6 Reopening a Window

To reopen a window of a participant in a conference, select *Participants* from the Menu bar. You will see a list of the conference participants with a box next to each name. An X in the box indicates they are not sending, so you cannot open their window on your screen. Select a name from this list without an X next to it to open their window on your screen.

5.8.7 Saving the Window Positions

To save the current positions of all the windows, select *File* from the menu bar and choose *Save Window Positions.* Window positions are saved according to the order they appeared, i.e. it saves where the second window to appear will be, etc...

6. Problems using CU-SeeMe

6.1 Error Messages

6.1.1 Startup Errors

"I found a digitizer component, but am unable to digitize. Continue in receive-only mode?"

This error is generally fixable by reducing the number of colors in the Monitor Control Panel.

SPBOpenDevice Failed. err= -227

This is caused by the sampling rate in Sound Out in the Sound Control Panel being set too high. Lowering the sampling rate should alleviate the problem.

VITA

Michael Bibeau is currently a First Lieutenant in the U.S. Air Force. He graduated in 1991 from the U.S. Air Force Academy with a Bachelor of Science in Computer Science. He then attended Undergraduate Pilot Training at Williams Air Force Base in Chandler, AZ where he earned his pilot wings for the Air Force. Upon completion of his Master of Science in Computer Science at Virginia Polytechnic and State Unversity, he will be returning to flying duty at Tyndall Air Force Base in Florida.

Michael J. Bibeau, 1Lt, USAF.