A MULTI-USER VIRTUAL ENVIRONMENT WITH EXTENSIBLE USER-FRIENDLY WEB-BASED INTERFACES

J. Mark Pullen
Department of Computer Science and C3I Center
George Mason University
Fairfax, VA 22030
mpullen@gmu.edu

Michael H. Nah
Lockheed Martin Technology Services Group
Crystal Gateway 2
1225 Jefferson Davis Highway Suite 400
Arlington, VA 22202
michael.h.nah@lmco.com

KEYWORDS: virtual environments, synchronous collaboration technology, computer-aided education, multimedia interfaces

ABSTRACT

Multi-User Virtual Environments (MUVEs) provide stateful environments with software objects having attributes and functions representing real-life objects in an object-oriented paradigm. The attributes of these software objects can be modified by multiple users, each stationed at geographically distant locations in the world, and passed on from one user to another in the common virtual environment retaining the modifications that were made by the first user. Participants in the virtual environment can change their perspectives by navigating between the various rooms that are available. We are developing the Distributed Integrated Internet Conferencing Environment (DIICE), a multi-user integrated collaborative conferencing environment that bridges the needs of the statefulness of the MUVE and the wide range of synchronous/asynchronous communications functionalities that are useful in collaborative applications. In this paper we discuss the design of DIICE from the standpoint of functions needed to enhance the MUVE for effective multimedia group communications. We have sought simple, user-friendly interfaces using a standard paradigm (the Web) to support, as far as possible, re-use of existing collaboration tools in the MUVE. Users have available to them a well-organized set of applications on the control panel of DIICE, which can be automatically launched on the desktops of all of the conference participants by the floor holder leading the discussion. DIICE was implemented in a client/server paradigm with Java applets. The control panel paradigm of DIICE provides an intuitive feel to the users, placing in a consistent location all information needed to sustain collaboration.

INTRODUCTION

The Internet has opened tremendous possibilities for distance education, where teachers and students can be distributed worldwide. However, most of today's Internet-based teaching is limited to "distance learning", based on student access to Web-based multimedia documents and using network-accessed multimedia to expand on an old paradigm, the correspondence course. We are seeking a new paradigm for learning and other collaborative work, based on the Multi-user Virtual Environment (MUVE). In this computer-mediated, network-based virtual world, the teacher can serve as mentor, answer student questions at the most "teachable moment", and provide up-to-date information that is missing from the "canned" version of the course. Even better, the MUVE supports collaborative learning, where students learn by interaction with each other as well as the teacher. The key to these capabilities is synchronous collaboration over networks, where the key ability is that of the collaborators to reach synergistic results by stimulating each other’s thought processes in real time.

At George Mason University (GMU) we have been investigating potential tools for synchronous distributed education. In a previous paper we reviewed potential values of synchronous distance education and showed how these can be met by a multimedia MUVE. Here we show how a collection of existing, Web-based collaboration tools can dramatically enhance the power of the MUVE for collaborative learning and other collaborative work.

The MUVE is not so much a simulation as a user interface environment within which simulations are presented. However, the form of presentation turns out to be critical to effective teaching. Therefore the enhancements to learning and other collaborative work we have investigated are very important to effective application of simulation.
MUVES AS COLLABORATION ENVIRONMENTS

Distance Education in the MUVE

We believe there is great value in collaborative learning styles, deriving from the psychological support for learning that is provided by group participation. The earlier paper (Pullen and Norris, 1998) provided a detailed description of synchronous learning styles that can be supported by a MUVE: lecturing, seminar, coaching, guided discovery, and unguided discovery. Each of these has its own place in effective teaching and learning, and all can be supported in a Web-linked multimedia MUVE. We further detailed the power of multimedia, particularly audio combined with graphics, to support this process. We noted that video, often the preconceived solution to distance education problems, is in general both the most expensive and least effective solution to most of these problems.

We find that, in general, networked multimedia technologies that support distance education support collaborative meetings with equal effectiveness (Pullen, 1998). Thus we have been able to support both teaching C++ to students at a high school in Italy, and coordinating the educational project with their teachers. The key to this broad applicability is the relative lack of structure in the MUVE, which allows it to support a variety of communication styles. In DICE we have sought to extend this unstructured environment with a clean, intuitive interface that makes maximum use of the Web paradigm as both a control interface and a framework to integrate existing collaboration tools.

Experiences with Multi-Media MUVEs

Until recently, personal interactions in a MUVE were limited to what could be accomplished with a primitive text interface, through which people typed messages to one another in real time. The messages were limited to one logical line, perhaps 2-3 lines on a computer screen, with excess text being lost. The text interface, which is still the basic communications medium on MUVEs, is cumbersome to use, in that it places a large cognitive burden on each user, who for example must know if he is “shouting,” “whispering,” “emoting,” or “paging.” He also must know whether there are factors affecting communication that are attributable to characteristics of the room currently occupied by his character. Typing errors are the cause of gibberish and misunderstandings requiring repeats, and a conversation among more than two people results in one's thread being interlaced with others'. One is never sure whether lack of response is a network problem or the result of distraction on the part of one of the other people. Humor, quick asides, flashes of wit can all increase rather than reduce noise in the communication between people. And, of course, there is no recourse to the adage that one graphic (let alone a moving picture) is worth a thousand bytes.

Our work with one MUVE technology, Lambda MOO (see http://lucien.berkeley.edu/moo.html), has extended and considerably eased the process of communication in two useful ways. Our early efforts in this vein were relatively simple, yet effective. We integrated into the MUVE the File Utilities Package, (FUP), to permit a teacher to display text files to all students in a room simultaneously, and we added real-time voice communications using CoolTalk, a widely available voice-over-Internet program. The positive reaction of teachers and students to these additions led us to believe that larger efforts in multimedia interfaces would be worthwhile. We were also encouraged by the success of a more esoteric multimedia interface created for the MUVE: Wyndhaven, created by Intermetrics Corporation (see http://www.inmet.com/caeti/wyndhaven/), provides visual and audio cues for a fantasy environment. Our approach has been more mundane, using multimedia for diagrams and spoken discussion. Another Web-based collaboration and teaching system, TANGO, has many attributes in common with DIICE but is not integrated with a MUVE (see http://trurl.npac.syr.edu/tango/).

We must add here that, although we value the MUVE as a way to provide the various instructional formats above in distance education, we also believe that, in or out of a MUVE, the most effective learning occurs when students undertake projects to use what they learn. In both distance education and MUVE-based collaboration, the key to success is involvement of the participant.

Our initial Web-based extension to the MUVE was BioGate, which permits access to a Lambda MOO through a Web browser, supports graphical point-and-click browsing of the MUVE, and is VRML-sensitive, allowing live interaction with graphical objects. In this environment we offered both a C++ course for high school computer teachers, and a C++ laboratory for high school students, in both cases working with overseas schools for dependents of U.S. military personnel. This work convinced us that the potential of MUVEs to address the age-old problem of how best to teach and learn is huge. But it also convinced us that MUVEs will need sophisticated multimedia interfaces in order to exploit the full power of networked collaboration, and that these interfaces will need to be as intuitive and user-friendly as possible in order to win acceptance by most MUVE users. We are actively pursuing one clear thread of opportunity based on these convictions: multimedia Web browser interfaces. We have superseded the BioGate interface with DICE, permitting capabilities for directed browsing and audio conferences, video conferences, and remote manipulation of workstation desktops. The advantages of the Web interface for this work are
significant: it supports a wide range of multimedia interfaces, provides access to many existing tools, and is already familiar to the majority of Internet users.

ARCHITECTURE OF DIICE

DIICE uses a client/server paradigm to allow multiple users to communicate with other participants in a session. The client is written in Java and runs in a Java-enabled Web browser, while the server is written in C++ and runs on a Unix platform. A DIICE session is integrated with the shared virtual space in the MUVE. Participants can interact in the MUVE, while they can also launch Web-based collaboration tools from the DIICE control panel. Within a given session, the clients interact by sending control messages or data to the server. The DIICE server forwards the messages to other clients participating in the session, or interacts with various collaboration tool servers to process the message. It then sends a response back to the originating client, and directions to other clients in the session as needed.

The floor management scheme used in DIICE allows clients to alternate in the role of floor holder for the session. The floor holder client has the ability to distribute information synchronously to other participants using the multimedia features of DIICE, and to launch Web-based applications through the other clients. Available applications include one-to-many voice communication and directed browsing (with an option for a real time slide show of Web-accessible multimedia documents). The client and server use the Internet Protocol Suite to support distributed interaction among dispersed participants over the Internet.

A major architectural distinction in DIICE is the difference between one-to-many (point-to-multipoint) applications and one-to-one (point-to-point) applications. The former are more desirable for DIICE in that they support a model of group collaboration; however in some cases (notably the Timbuktu desktop sharing application) the latter provide functions that are not available for groups larger than two. The DIICE floor control paradigm is based on one-to-many communication. It supports many-to-many collaboration by being sufficiently agile that the floor can be passed from user to user in under one second. This allows the dynamic of a multiparty conversation to be sustained while avoiding the complexity of many-to-many flow control.

CURRENT DIICE APPLICATIONS

One-to-Many Applications

Webference is an applications tool that we have created as an integral part of DIICE. As its hybrid name suggests, it is a conferencing application for distribution of web pages. The floor holder can navigate around the web using the Webference in two different ways: by entering URLs in a window, or by clicking on hyperlinks present in the current web page. As the floor holder navigates around the web, each of the web pages visited is displayed synchronously to the other participants within the same session. The history of URLs visited is displayed as a list on the Webference interface allowing users to revisit the desired URLs by clicking on a list entry. A particularly useful feature of the Webference is the slideshow option for presenting Web slides at a specified time interval. The slides can be created by visiting URLs, then saving the URL list.

SpeakFreely is a freeware package that supports Internet telephony (see http://www.fourmilab.ch/#speakfree) and has been used effectively in distance education (Pullen and Benson, 1998). The DIICE server uses a modified SpeakFreely echo server to replicate audio packets sent by the floor holder and distributes them to the other SpeakFreely users within the session, thus providing one-to-many voice transmission. In accordance with the DIICE floor management paradigm, only the floor holder can send audio. Therefore any audio packets sent to the DIICE server by a participant not holding the floor are neither replicated nor forwarded to the other users.

Whiteboard. A multi-user whiteboard application is also available in DIICE providing a capability to collaborate on a graphics project or even a text document in a synchronous fashion. Consistent with the floor management scheme, modifications on the shared whiteboard can be made only after becoming a floor holder of the session.

Mbone. The Internet multicast backbone (Mbone) multimedia tools (Macedonia and Brutzman, 1994) are available under DIICE. The full capabilities of Mbone’s audio, video and whiteboard tools are available. The addition of Mbone provides an integrated video and audio communication for situations in which observing video can convey needed information. An example could be a board meeting between directors of a company.

Point-To-Point Applications

With point-to-point tools, the floor holder is required to select the name of the user with whom communication is desired. The specified application is launched automatically by the DIICE client at the remote site of the person chosen.

CoolTalk is Internet telephony software from Netscape that provides a one-to-one connection for private conversations that do not use the SpeakFreely echo server. It supports private voice conversations within a DIICE session.
Timbuktu is a commercial product for desktop sharing, whereby a user of one system can observe or control via the Internet the desktop of another, remote user. Launched by DIICE, this capability can play an important role for demonstration and assistance with operation of software. Used with a voice communication feature (CoolTalk or SpeakFreely) to instruct or explain uncharacteristic events on the remote desktop, DIICE with Timbuktu shows the results of a program running locally to a remote user.

DIICE SYSTEM DESIGN

Client Interface

The initial DIICE login interface consists of a Java applet with input fields for user authentication embedded on the left frame, complemented with a DIICE user tutorial page on the right frame. The design assists the new users of DIICE in becoming familiar with the initial setup and configuration of the application tools. It also provides descriptions of functions provided by the integrated tools available in DIICE before login. After the user submits the login, the left frame is replaced with the DIICE control panel and the right frame by a Java telnet interface to the MUVE server (figure 1). The MUVE can be any system accessible via telnet; we use the GMU version of Lambda-MOO, which provides capabilities to wander from room to room and manipulate and modify the existing virtual objects in the virtual environment. In addition to spatial context and persistent objects, the MUVE provides synchronous textual communication with the other users in the DIICE session.

The control panel located on the left frame of the client features at the top a red “exit session” button as well as a current speaker field indicating the current floor holder for the session. In the center of the left panel a list of users currently active in the session is displayed. Below this is a message text area used to return DIICE server messages to the user. Below this are buttons to select the floor control rules (see below). At the bottom of the panel are buttons associated with each available application, as configured for the current user. When an application is chosen by the floor holder, the list of users who are active in the session and have the particular application configured are displayed on a pop-up list on the control panel. Using these, the floor holder can launch the one-to-many applications launched through all participating clients. A JavaScript message also alerts the floor holder if there are active users who do not have the particular application being launched configured in their client. Questions concerning the features of the control panel can be answered by clicking on the ‘help’ button.

Floor Management

The DIICE control panel provides an efficient floor management scheme and with it a method of providing multiple users within a session to alternatively become the leader or floor holder of a session and distribute local information synchronously to others in the session. To request the floor, a participant needs to click on the “REQUEST FLOOR” button. This causes the floor control rule (ALWAYS, NEVER, or SOMETIMES) set by the current floor holder to be invoked. If this option is set to SOMETIMES, the floor is not immediately given to the requested participant. Instead, the name of the participant(s) requesting for the floor is highlighted on the list of currently active users. The floor holder clicks on the name of the requester to transfer the floor.

Client Design

The DIICE client was designed with dynamic integration of Web-based tools in mind. The client software has a highly modular design, which supports integrating additional applications to the DIICE client’s control panel. A Microsoft mSQL database maintains configuration, authentication, and status information for all DIICE users. The configuration information is entered through a Web/forms interface that collects all needed information.
The configuration database is used dynamically to generate application buttons on the control panel reflecting the type of session or conference of interest.

The DIICE client software is implemented as a Java applet embedded within an HTML page partitioned with frames, allowing platform independence and ease of configuration. The DIICE user is initially presented with a login applet with input fields for login name, password, and the MUVE meeting room to enter (there is a different session for each room). The login name and password pair is used to query a matching user in the existing DIICE user database. If the login/password pair matches an entry in the database, the client generates one “click-on” button for each application available to the user. Clicking on this button will launch the application. When the floor holder launches an application, all participating clients also will launch the same application automatically.

Server Design

The DIICE server is a concurrent, multi-threaded server implemented in C++ and designed to accommodate multiple sessions simultaneously. Each session is independent of other sessions and represents a specific location or room in the MUVE. The server handles messages from clients according to the status of the client. If the client is the floor holder of a session, the messages are sent to all of the other participating clients within the same session. This allows a floor holder of the session to lead and present to the rest of the group.

The DIICE server is object-oriented and uses lists to maintain status of all connected sessions. It consists of multiple threads each handling messages for the main, verification, control panel, Webference, slideshow, and the SpeakFreely echo servers. The server also interacts with an mSQL user database to obtain user authentication and user application configuration information. It uses an open-ended TCP/IP connection management scheme that can handle large numbers of sessions and clients. Clients are identified uniquely and authenticated by their Internet addresses in addition to user login.

Each application in a DIICE session is represented with a session profile marking the clients that have the applications currently opened on their local machines. The server control panel and Webference threads wait for arrival of messages from clients and parse the protocol ids from the received messages. Messages are processed according to a protocol identifier established between the client and server. When the floor holder distributes messages using one of these applications, the session profile for the application is consulted to determine to which clients the messages should be sent.

Part of the DIICE server is a modified SpeakFreely echo server consisting of two UDP sockets, one for the control message and the other for data. This server extends the point-to-point SpeakFreely tool to a point-to-multipoint capability. If a control message is received, after checking the validity of the source IP address the client is added to the client list. If a data message is received, determination is made as to whether it originated from the floor holder, in which case the message is sent to all of the other participants in the session and ignored otherwise.

The DIICE server handles the disconnection of the control panel and Webference applications appropriately according to the status of the client. If the disconnecting client is a floor holder, the entire session is closed; if another client closed, only that client departs the session.

Database Integration

DIICE provides authentication and configuration services by connecting to a DIICE user database storing these attributes of each user. The attributes are established through a separate configuration webpage. They are:

```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>login_name</td>
</tr>
<tr>
<td>Password</td>
<td>first_name</td>
</tr>
<tr>
<td>last_name</td>
<td>city</td>
</tr>
<tr>
<td>country</td>
<td>e_mail</td>
</tr>
<tr>
<td>ip_address</td>
<td></td>
</tr>
<tr>
<td>configuration flags</td>
<td>for all available tools</td>
</tr>
</tbody>
</table>
```

If a user attempting to login exists in the current user database, the user is sent an authentication message containing the list of applications that was registered as having been configured on the particular user’s system. The DIICE server uses this database to determine the list of users among the active participants in the specific session to have the particular application configured on their local systems.

MUVE Integration

The combination of DIICE and MUVE gives users access not only to the collaborative features of DIICE but to the statefulness and the object-oriented approach of the MUVE. Objects within the virtual environment (e.g. a conference room) provide the virtual setting for the collaborative features available in DIICE. Modifications made in the current login session to objects within this computer room will persist until the objects are changed again in subsequent sessions. In addition to the dynamic creation of the launch buttons associated with the configured applications on the control panel, the user is logged onto a virtual room in the MUVE server specified during DIICE login. Within this room objects may exist,
appropriate for the purposes of the group that uses the room. Roaming from one MUVE room to another changes not only the virtual setting in the MUVE, but also updates the DIICE session to the new session (and room) and updates the list of users on the DIICE client to that of the new session. The DIICE session and the MUVE room refer to the same virtual space.

CONCLUSIONS AND FUTURE WORK

We have described DIICE, a Web-based multimedia client-server system that has been developed and integrated with a MUVE at GMU. This work was motivated by our experiences in using text-based MUVEs, which indicated that multimedia tools could greatly improve the quality of the collaboration environment. We chose the Web paradigm because of its flexibility and the richness of emerging Web-based collaboration applications. We have created a carefully crafted Java-based client and C++-based server that can support multiple simultaneous sessions organized into MUVE rooms. Our server works in conjunction with a database server, MUVE server, and servers for other multi-user tools. Both client and server are designed to be as robust and user-friendly as possible and also to be easy to extend.

In the future we intend to pursue extensions of the Web-based interface, for example a Virtual Reality Modeling Language (VRML) interface into three-dimensional virtual simulations that will be shared with the Naval Postgraduate School’s Distributed Interactive Simulation/Java/VRML project (see http:// www.stl.nps.navy.mil/dis-java-vrml/). Ultimately, we would like to create the ability for VRML objects to be more fully integrated into the MUVE itself. VRML and other emerging technologies offer the possibility of dramatically enhanced collaborative experiences. The other critical aspect of our work is expanding the range of experience with users of the DIICE-interfaced MUVE. Whereas most usage to date has been in the laboratory, we are currently seeking expanded use by a friendly but skeptical community, perhaps a group of the teachers who provided our initial MUVE-based experience in distributed education. We believe we will learn a great deal from these users that can help to shape continued growth of networked multimedia collaboration.

REFERENCES

Macedonia, M. and D. Brutzman, “Mbone Provides Audio and Video Across the Internet”, IEEE Computer, April 1994


Pullen, J., “Synchronous Distance Education and the Internet”, Internet Society Annual Conference, Geneva, Switzerland, July 1998


ABOUT THE AUTHORS

J. Mark Pullen is an Associate Professor of Computer Science and a member of the C’I Center at George Mason University, where he heads the Networking and Simulation Laboratory. He holds BS and MS degrees in Electrical Engineering from West Virginia University, and the Doctor of Science in Computer Science from the George Washington University. He is a licensed Professional Engineer and a Fellow of the IEEE. Dr. Pullen teaches courses in computer networking, and has active research in networking for distributed virtual simulation and networked multimedia tools for distributed education. Dr. Pullen recently received the IEEE’s Harry Diamond Memorial Award for his work in networking for distributed simulation.

Michael H. Nah is a Senior Software Engineer with Lockheed-Martin Technology Services Group, where he works in collaboration technology. He holds the BS degree in Aerospace Engineering from the University of Virginia, MS degree in Mechanical Engineering from Boston University, and MS degree in Computer Science from George Mason University, where he currently is a Ph.D. candidate.