

Purdue University

Purdue e-Pubs

Department of Computer Science Technical
Reports

Department of Computer Science

1993

Collaborative Multimedia Game Environments

Vinod Anupam

Chandrajit L. Bajaj

Report Number:

93-086

Anupam, Vinod and Bajaj, Chandrajit L., "Collaborative Multimedia Game Environments" (1993).
Department of Computer Science Technical Reports. Paper 1099.
<https://docs.lib.purdue.edu/cstech/1099>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

**COLLABORATIVE MULTIMEDIA
GAME ENVIRONMENTS**

**Vinod Anupam
Chandrajit L. Bajaj**

**CSD-TR-93-086
December 1993**

Collaborative Multimedia Game Environments*

Vinod Anupam Chandrajit L. Bajaj

Department of Computer Science,
Purdue University,
West Lafayette, IN 47907
{anupam,bajaj}@cs.purdue.edu
Tel: 317-494-6531, FAX: 317-494-0739

Abstract

We have implemented a multi-user distributed and collaborative environment, SHASTRA, on the multimedia desktop. This application conferencing substrate facilitates user level cooperation and supports many modes of asynchronous and synchronous multiple-user interaction. In this paper we describe how the SHASTRA infrastructure can be used to implement realistic virtual environments for collaborative interaction for entertainment and instruction. We present examples of collaborative multimedia games we have built in the SHASTRA environment.

Keywords:

CSCW ; Groupware; Multi-Player Games; Virtual Team Support; Multimedia Communication; Synchronous Computer Conferencing; Virtual Environments;

1 Introduction

Advances in computer technology and high speed networking, in conjunction with efficient compression techniques and audio-video processor architectures have made high speed full color graphics, high resolution motion video, and CD quality audio readily available on the desktop. Harnessing this low-cost high-speed workstation technology and utilizing their multimedia functionality and performance has made it feasible to explore the next generation of entertainment and educational facilities in a virtual world – collaborative multimedia games.

*Supported in part by NSF grants CCR 92-22467, DMS 91-01424, AFOSR grant F49620-93-10138, NASA grant NAG-1-1473 and a gift from AT & T

2 The State of the Art

There are a variety of games available on computers nowadays. In general, Personal Computers and non-graphics workstations use bitmapped *2D* graphics. Most games supported on the X window system, for example, fall into this category. Graphics workstations use *3D* graphics, but in general the demands of interactivity cause them to compromise on the aspect of graphics realism. Occasionally, the software uses available audio facilities. The quality of these games is currently significantly inferior to that of popular arcade games. We are using the SHASTRA infrastructure to build advanced collaborative environments. This includes games where participants can interact with one another as well as with virtual players; multi-person simulators with audio-video communication facilities in addition to the shared environment; and collaborative instructional and educational environments.

The games domain can be classified based on the following criteria. The taxonomy almost directly translates to platform requirements at the hardware or software level.

- Graphics and Realism
- Interaction Rate
- Multimedia Usage:

At the low end of the Graphics and Realism domain exist ASCII interface based games. Most popular games (including arcade games) use *2D* bitmapped graphics at the user interface. Platforms for this mode of graphics are widely available, and many sophisticated games have been created and deployed. *3D* graphics is more demanding in terms of computational power of the platform, but provides a means of expressing much more realism than the *2D* bitmapped mode. Many current

graphics workstations support fairly sophisticated 3D games. At the high end of the graphics and realism domain, texture mapped free form surface based 3D graphics can be used to build very realistic gaming scenarios. Needless to say, this requires significantly more computation power in the machines. This need will be met by the multimedia workstations of today. The graphics requirements of a game are inherent to it. Word games minimally require a textual interface, and card games minimally require a bitmapped color interface. Our attempt at simulating realism automatically forces us into the 3D domain.

Interaction rate, once again, is intrinsic to the application. Computation in word and board games, and others in that class, is usually completely player driven. The software responds only to user input, and the machine is idle most of the time. Continuous feedback based games use monitored valuations and state information to continuously respond to user input. Interaction speed in these cases is typically a function of the amount of computation required to maintain the user interface. Most current computers are powerful enough to support high interaction rates owing to their high speed. However, the quest for realism imposes computational overhead which can easily exceed the capability of even the most powerful computers. The saving grace is that visual simulation can often be used to project a high degree of realism. Faster processors will make it possible to include physical simulation to more accurately convey the notion of a virtual world.

Current arcade games already exploit multimedia technology for synthetic and recorded audio playback, in conjunction with high speed bitmapped graphics, to create a sensory ambience for the game participant in the virtual environment. Most other computer based games rarely exploit the multimedia facilities of today's high performance machines. Current and evolving technology has enabled the inclusion of high quality motion video, both live and recorded, into virtual environments.

3 Requirements of the Collaborative Environment

Virtual Environments for entertainment in a distributed and collaborative setting minimally require an infrastructure that supports the following.

1. Distribution facilities that will permit users to participate in the interaction across a network. This includes Directory facilities that identify the address of the virtual environment, Location facilities that indicate how and where it can be connected to, and Connection and Distribution facilities to actually provide and support a network connection to the environment.
2. Session Management in a multi-user environment to deal with the low level details of multi-party interaction – shared data access, concurrency control, constraint management etc. Participants in the world may be real (other players) or synthetic (other programs).
3. Multimedia Communication facilities to support online audio and video communication channels inside or outside the context of the game, to enhance the quality of shared interaction.
4. High Speed Textured 3D graphics for visual realism in the shared environment, since the real world is three dimensional
5. Modelling support for virtual realism which includes both graphical and physical modelling to be able to conveniently model the virtual world.

In this context, certain design decisions also impact infrastructural requirements. Some games may not require powerful graphics facilities to support the notion of a shared virtual world. In others the user would just see the results of interaction of all the participants, without actually 'seeing' them – Effectively, their actions would interact and they would share views of the virtual world. At the other extreme in a shared multi-user environment, users would 'see' others navigating

the virtual world in some form, and interact with their remote presence. Graphics performance and realism, as well as multimedia communication quality, would be a big concern in this case.

4 The SHASTRA Infrastructure

The SHASTRA environment consists of a group of interacting applications [1]. Some applications are responsible for managing the distributed environment (the Kernel applications), others are responsible for maintaining collaborative sessions (the Session Managers). Some applications provide specific communication services (the Service Applications), and others provide scientific design and manipulation functionality (the SHASTRA Toolkits). Another set of applications implement games and interaction environments. Service applications are special purpose tools for multimedia support – providing mechanisms of textual, graphical, audio and video rendition and communication.

Different tools register with the environment at startup providing information about what kind of services they offer (Directory), and how and where they can be contacted for those services (Location). The environment provides mechanisms to create remote instances of applications and connect to them in client-server mode (Distribution). In addition, the environment provides support for a variety of multi-user interactions (Collaboration). It provides mechanisms for starting and terminating collaborative sessions, and joining or leaving them.

A permissions based regulatory subsystem permits control of data flow at runtime, providing a variety of interaction modes. Collaboration in SHASTRA can occur in the REGULATED (Turn-taking or Master-Slave) mode or in the UNREGULATED (Free Interaction) mode. In the REGULATED mode, users take turns by passing a baton. The collaboration infrastructure of SHASTRA has a two tiered permissions based regulatory subsystem used to control interaction

primarily in the UNREGULATED mode. Permissions are specified on a per-site as well as a per-object basis. Site permissions control the kind of interaction a participant can have in the collaboration context. Object permissions control what actions the participant can take on the shared objects. In general, the more restrictive of Site and Object permissions applies for shared objects. SHASTRA permissions control 'Access' to a view of the conference, local viewing controls to 'Browse' a view, rights to 'Modify' conference state, rights to 'Copy' shared objects, and rights to 'Grant' permissions to other users.

Permissions are maintained in the Session Manager. Site permissions are regulated by the group leader or anyone with the *Grant* permission. Object permissions are regulated by the owners of shared collaboration objects. Different permissions settings for participants and objects generate a variety of interaction modes at runtime. *E.g.* Giving all but one participant only *Access* permission and one participant *Modify* permission, effectively simulates a Master-Slave situation, where one participant alters the state of the collaboration, and everyone else observes the results. Changing everybody else's permission to *Browse* results in a flexible Master-Slave situation, with every site capable of independent local views. Allowing everybody to *Modify* the state of the collaboration creates a free interaction situation. Similarly, setting object permissions regulates the operations a participant can perform. Specifying *Access*-like permissions for a subset of the collaborating group, and denying it to the others effectively supports private object exchange.

Current functionality in SHASTRA supports simultaneous but independent, unsynchronized virtual channels for transmission of text, graphics, images, audio and video information over the network in multiple-site settings.

We have used the collaboration support infrastructure of SHASTRA to build different interaction scenarios. SHASTRA provides a distribution mechanism, generic session management facilities

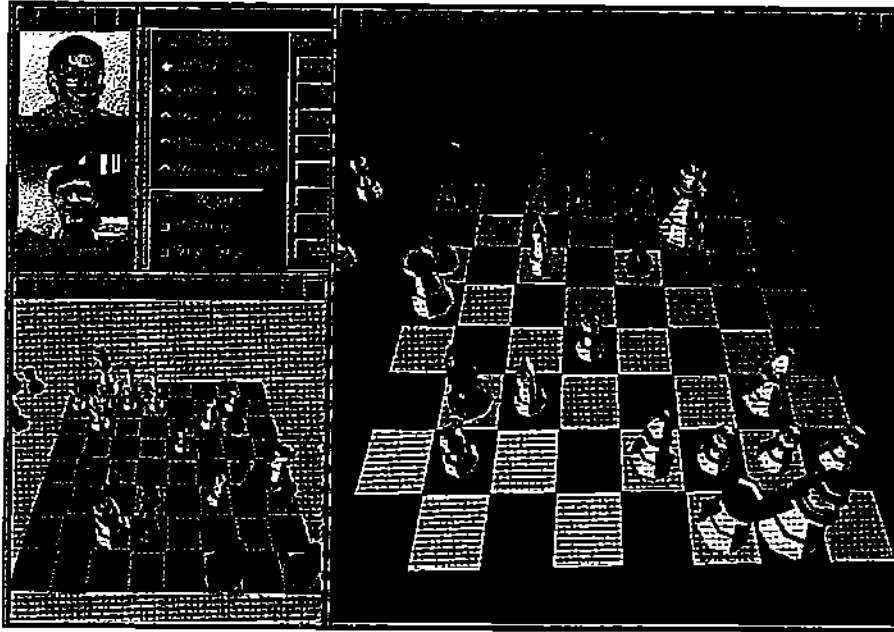


Figure 1: SHA-CHESS – Collaborative 3D Chess

and multimedia communication support in a distributed setting, as well as flexible dynamic control of interaction modes. From the perspective of multimedia interaction support, SHASTRA provides facilities for recording and playback of captured and synthetic information in the virtual world, as well as mechanisms for integrating such facilities into applications. To augment and enhance collaborative interaction, SHASTRA can be used to provide multimedia communication support with real and synthetic players in the virtual world.

5 Example Virtual Environments

SHA-CHESS supports a shared virtual 3D chessboard and typifies virtual collaborative environments for games and entertainment-oriented interaction. SHA-CHESS is a collaborative chess substrate in the SHASTRA environment. As a stand-alone application, it provides a 3D graphical interface on which chess games can be played. It is built on top of XS, a hardware independent

3D graphics system. See Figure 1. SHA-CHESS lets a user play against a chess playing program, or against another user, locally. It supports a regulated mode where the system allows only legal moves, in his turn, as well as an unregulated mode, where the system just provides a game playing substrate without regulating interaction, like a physical chess board.

A collaboration of SHA-CHESS instances creates a virtual world and provides an interface that lets a group of geographically separated chess players synchronously interact over a shared virtual chessboard. The Session Manager causes all the participating SHA-CHESS instances to create a shared window where the chess game is played by users across the network. SHA-CHESS exploits the permissions mechanism of SHASTRA to support a variety of interaction modes in which the multiple users interact in the virtual environment. The group leader (or any user with the Grant permission) can configure the constraint management subsystem of the Session Manager, as well as the permissions of the different players, to regulate the board in many ways.

At one extreme, the Session Manager performs no regulation. It simply transmits moves made by different players who have Modify permission and updates the view at all sites with Access permission. Using audio, video and text communication channels to coordinate matters, users can play a game successfully in this mode. Alternately, the group leader can give exactly two people Modify permission for the session, and they would be the only active participants, with everyone else getting a current view of the board. If the Session Manager is also put in regulated mode, allowing only legal moves in turn, a tournament situation is simulated in this virtual environment. The group leader can direct the Session Manager to divide the group of individuals into teams such that any member of a team can make a move for that team. In yet another scenario, the group leader can switch to Master-Slave mode with unregulated moves to use the SHA-CHESS collaboration as a blackboard to teach a group of individuals fundamentals of the game of chess, or

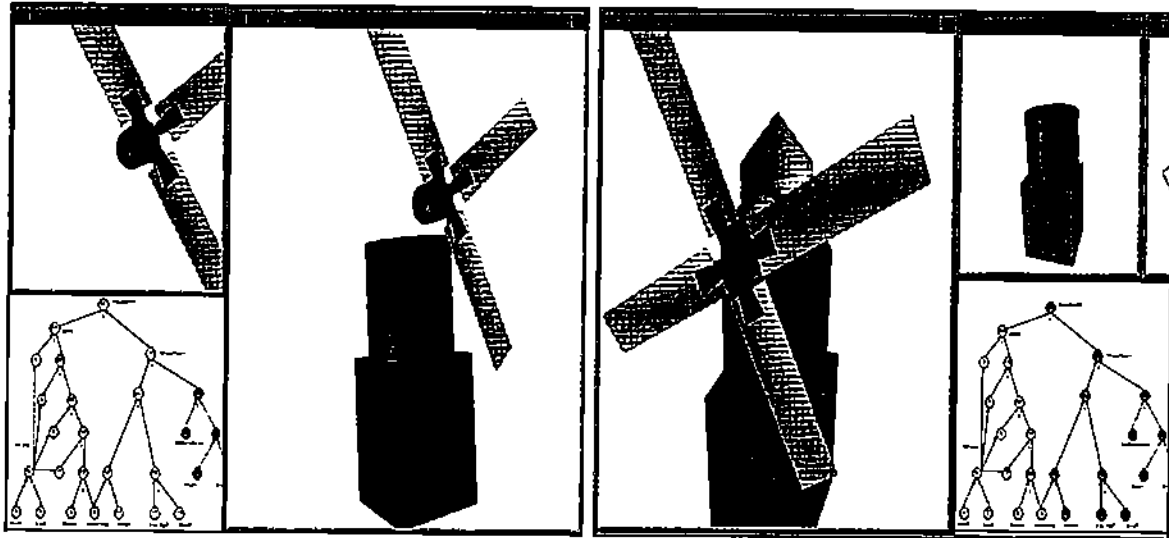


Figure 2: SHA-LEGO – Collaborative 3D Design

to discuss strategy. Multimedia communication facilities provide a rich interaction environment.

SHA-LEGO supports a shared virtual 3D design environment and typifies virtual collaborative environments for “building block” games and other entertainment-oriented as well as educational interaction. A collaboration of SHA-LEGO instances creates a virtual world and provides an interface that lets a group of players synchronously interact over a shared design task. Every participating SHA-LEGO session creates two shared windows in which all the cooperative interaction occurs. More local windows can be created if desired. One shared window contains the design graph that is used to regulate the entire operation. The second window contains models as they are introduced to or created in the session. Users introduce leaf node objects into the session by selecting them into the Collaboration Window. The Session Manager is responsible for providing access to the objects at all participating sites which have the Access permission, and for permitting interaction relevant to the operation at sites which have Modify permission for the collaboration. The left part of Figure 2 shows one site in the building of a simple windmill block model. The right part of Figure 2 shows another site at the end of the operation.

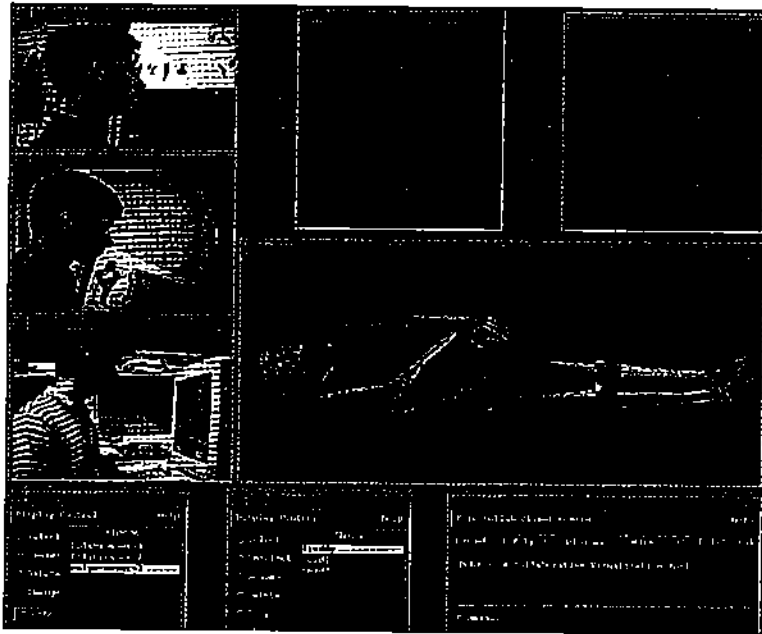


Figure 3: SHA-VAID – Collaborative Medical Modelling and Visualization

SHA-VAID supports a shared medical modelling and visualization environment and typifies virtual collaborative environments for education-oriented interaction. A collaboration of SHA-VAID instances creates a virtual world and provides an interface that supports collaborative modelling and visualization, allowing multiple users to share and interact over extremely large volume data sets while viewing multiple iso-surfaces and renderings with independent viewing directions, cutaways and shading parameters. The modelling and rendering algorithms uses the computational power of multiple networked workstations to speedily produce piecewise trivariate interpolants (modelling) and translucent shaded images of isosurfaces as well as direct volume rendering of extremely large volume data sets (visualization). Figure 3 shows one of two sites with independent private windows, and a shared conference window.

6 Computation Model

SHASTRA Conferences are implemented in a hybrid Replicated-Centralized computation model. A Central Session Manager regulates activity of multiple instances of Front Ends. The method of operation can generally be termed as *Proxy* shared window management. The Session Manager has a Core part and an Application-Specific part. These parts communicate with a Collaboration Slave in the Front through their respective network interfaces. Front Ends have a Collaboration Slave which maintains shared context and performs collaboration-relevant operations under directives from the Session Manager.

6.1 Replicated Fronts

The multiple Front instances provide performance benefits, especially since most of the tasks we are interested in are expressed through 3D graphics. The replication scheme supports heterogeneity – the Session Manager communicates with Fronts at an abstract level, and isn't concerned with details of how the Front actually executes its directives. In a heterogeneous environment, this is a big win, because replication allows Fronts to execute on a variety of hardware platforms. Yet another benefit of the replication scheme is that it intrinsically supports the notions of Private and Shared workspace and interaction. Only activity in the collaboration context (windows and work areas) is shared between participants. Local windows isolate private interaction. Fronts provide regulated mechanisms for moving work between private and shared workspaces. The underlying notion is to provide as non-intrusive an environment as possible for collaborative interaction.

6.2 Hardware Independence

In a heterogeneous computing environment, SHASTRA applications achieve Hardware Independence by building on top of high level abstractions, above the greatest common denominator. We assume the availability of the X Window System (X11R5) for user interfaces.

For 3D graphics, we designed XS: A hardware-independent abstract graphics library. XS is a powerful mechanism for engineering user interfaces for scientific manipulation applications on hardware graphics systems. It consists of a suite of libraries which provide access to system-dependent graphics facilities in a uniform, system-independent manner. Each hardware graphics system supported is represented by a library in the suite. All libraries implement the same graphics paradigm and present the same function-call interface, permitting maintenance of source-level portability across several graphics systems in application programs. The XS suite includes libraries for SGI workstations (GL), X11 (Xlib), HP workstations (STARBASE) and IBM-compatible personal computers (Windows). We will extend XS to include new platforms when the standardization of 3D graphics platform interfaces becomes widely accepted.

Multimedia interfaces for audio and video present a similar problem, which we have resolved in a similar manner. Hardware and media interaction is isolated in the Fronts that deal with it directly. Platform independence is achieved by building applications atop abstract libraries which hide hardware specifics. Again, the abstract libraries can be easily extended to standardized interfaces as they evolve. We currently support audio interaction on Sun and SGI platforms. Video capture is presently limited to Sun workstations. Video playback is handled on top of X11.

7 Conclusion

The infrastructure of the SHASTRA collaborative environment provides mechanisms to support a variety of multi-user interactions spanning the range from demonstrations and walk-throughs to synchronous multi-user collaboration. The infrastructure also facilitates the exchange of multimedia information.

We have briefly described the architecture and runtime environment of SHASTRA, and have demonstrated how we have used it to build shared virtual environments in a collaborative setting. We hope to learn valuable lessons in computer mediated group activity and computer supported cooperative work from the system.

References

- [1] V. Anupam and C. Bajaj. SHASTRA - An Architecture for Development of Collaborative Applications. In *Proc. of the Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 155–166. IEEE Computer Society Press, 1993.