Sharing VLNET Worlds on the Web*

Daniel Thalmann[1], Christian Babski[1], Tolga Capin[1]
Nadia Magnenat Thalmann[2], Igor Sunday Pandzic[2]

[1]Computer Graphics Laboratory (LIG), Swiss Federal Institute of Technology (EPFL)
CH1015 Lausanne, Switzerland

[2]MIRALAB, University of Geneva
CH-1211 Geneva 4, Switzerland

Virtual environments define a new interface for networked multimedia applications. The sense of "presence" in the virtual environment is an important requirement for collaborative activities involving multiple remote users working with social interactions. Using virtual actors within the shared environment is a supporting tool for presence. In this paper, we present a shared virtual life network with virtual humans that provides a natural interface for collaborative working and we describe the bridge we realized between this 3D shared world and the Web through a system of 3D snapshots.

***Key Words**: Virtual Life, VRML, Networked, Virtual Actors*

1 Introduction

Increasing hardware and network performance together with the software technology make it possible to define more complex interfaces for networked multimedia applications. Using a Virtual Reality based environment is an increasingly popular method as an intuitive interface for this purpose. A networked Virtual Reality environment can provide a more natural shared environment, by supporting interactive human collaboration and integrating different media in real-time in a single 3D surrounding. Its strength comes from the fact that it supports awareness of and interaction with other users; and it provides an appropriate mechanism for interaction with the environment by supporting visual mechanisms for data sharing and protection.

Providing a behavioral realism is a significant requirement for systems that are based on human collaboration, such as Computer Supported Cooperative Work (CSCW) systems. Networked CSCW systems also require that the shared environment should: provide a comfortable interface for gestural communication, support awareness of other users in the environment, provide mechanisms for different modes of interaction (synchronous vs. asynchronous, allowing to work in different times in the same environment), supply mechanisms for customized tools for data visualization, protection and sharing.

Virtual Reality can provide a powerful mechanism for networked CSCW systems, by its nature of emphasizing the *presence* of the users in the virtual environment (VE). This can be accomplished through the support of:

- representing the users and special-purpose service programs by 3D virtual actors in the virtual environment,
- mechanisms for the users to interact with each other in the natural interface via facial interaction and body gestures of their virtual actors,
- mechanisms for the users to interact with the rest of the virtual environment through complex and realistic behaviors such as walking for navigation, grasping, etc.
- user-customized tools for editing the picked objects, depending on the object type (e.g. images, free-form surfaces)

There have been an increasing interest in the area of networked virtual environments recently [Zeltzer-Johnson94][Macedonia-Zyda94][Stanfield94][Gisi-Sacchi94]. These systems are centralized on the task level, and less work has been done on supporting the psychological aspects [Travis-Watson94][Rich-Waters94], including the sense of *presence* in the shared environment through gestural communication with other users or virtual agents and the representation of the whole body. The aim of VLNET (Virtual Life Network) is to provide a networked 3D environment that provides mechanisms for supporting the sense of presence, while integrating different media and virtual humans in the same virtual world.

The paper starts with the properties of the system: the environment, methods for modeling and animation of virtual actors in this environment, facial interaction support, communication model and further

*Published in Computer Networks and ISDN Systems, Elseveir, Vol. 29, 1997, pp. 1601-1610 improvements using autonomous actors. Then we discuss the implementation aspects of the system and we present the bridge we realized between VLNET and the Web. Finally, we present our concluding remarks and expectations for future improvements.

2 Properties of the System

Our system supports a networked shared virtual environment that allows multiple users to interact with each other and their surrounding in real time. The users are represented by 3D virtual human actors, which have similar appearance and behaviors with the real humans, to support the sense of presence of the users in the environment.

The environment incorporates different media; namely sound, 3D models, facial interaction among the users, images represented by textures mapped on 3D objects, and real-time movies. Instead of having different windows or applications for each medium, the environment integrates all tasks in a single 3D surrounding, therefore it provides a natural interface similar to the actual world. The environment works as a general-purpose stream, allowing the usage of various models for different applications.

In addition to user-guided agents, the environment can also be extended to include fully autonomous human agents which can be used as a friendly user interface to different services such as navigation. Virtual humans can also be used in order to represent the currently unavailable partners, allowing asynchronous cooperation between distant partners.

2.1 The Environment

The objects in the environment are classified into two groups: fixed (e.g. walls) or free (e.g. a chair). Only the free objects can be picked, moved and edited. This allows faster computations in database traversal for picking. In addition to the virtual actors representing users, the types of objects can be: simple polygonal objects, image texture-mapped polygons (e.g. to include three-dimensional documents, or images in the environment), etc. Once a user picks an object, he or she can edit the object. Each type of object has a user-customized program corresponding to the type of object, and this program is spawned if the user picks and requests to edit the object. Fig. 1. shows a general view of an example environment.

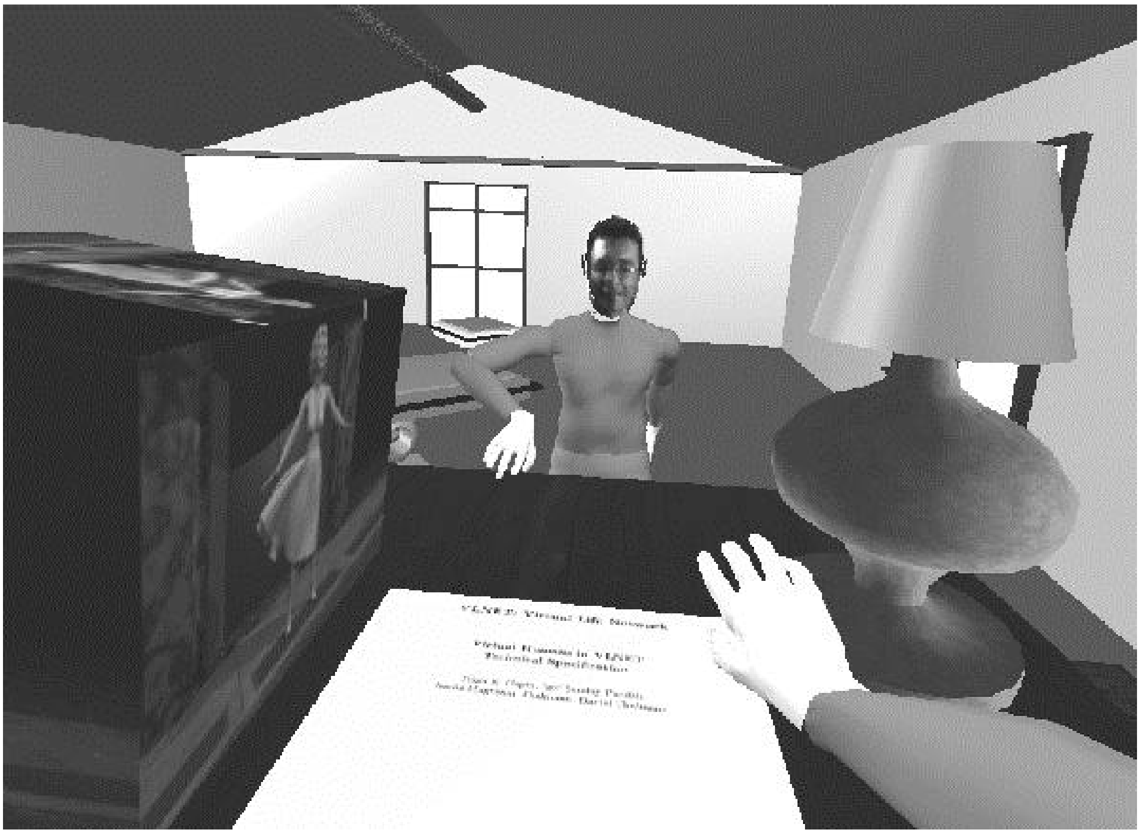

Fig. 1. The environment integrates different media in a single stream

2.2 Virtual Actors

It is not desirable to see solid-looking floating virtual actors in the environment; it is important to have motion control of the actors to have realistic behaviors. There are numerous methods for controlling motion of synthetic actors. A motion control method specifies how the actor is animated and can be classified according to the type of information it privileged in animating the synthetic actor.

The nature of the privileged information for the motion control of actors falls into three categories of motion control method:

- The first approach corresponds to methods heavily relied upon by the animator: rotoscopy, shape transformation, keyframe animation. Synthetic actors are *locally controlled* by the input of geometrical data for the motion.
- The second way is based on the methods of kinematics and dynamics. The input is the data corresponding to the complete definition of motion, in terms of forces, torques, constraints. The task of the animation system is to obtain the trajectories and velocities by solving equations of motions. Therefore, it can be said that the actor motions are *globally controlled*.
- The third type of animation is called behavioral animation and takes into account the relationship between each object and the other objects. The control of animation can also be performed at task-level, but one may also consider the actor as an *autonomous creature.* The behavioral motion control of the actor is provided by providing high-level directives indicating a specific behavior without any other stimulus.

Each category can be used for guiding virtual actors in the virtual environment, however it is important to provide appropriate interface for controlling the motion. In addition, no method alone is convenient to provide a comfortable interface to accomplish all the motions, therefore it is necessary to combine various techniques for different tasks.

For the current implementation, we use local methods for the users to guide their virtual actors for navigating in the virtual environment and picking objects using various input devices; and behavioral animation for realistic appearance based on these inputs and the behavioral parameters, such as walking for navigation and grasping for picking. This set of behaviors can easily be extended, however these behaviors are sufficient to perform everyday activities, providing minimum set of behaviors for teleconferencing.

The walking behavior is based on the Humanoid walking model, guided by the user interactively or automatically generated by a trajectory. This model includes kinematical personification depending on the individuality of the user [Boulic-Thalmann90]. Given the speed and the orientation of the virtual actor with the personification parameters, the walking module produces the movement in terms of the joint values of the articulated body.

The grasping behavior is also important in order to achieve realistic looking motions of the virtual actors. Although one could apply a physically correct method, our concern is more on the visual appearance of the grasping motion. The grasping motion is automated by the user giving directions on which object to grasp, and the virtual actor doing the appropriate grasping operation depending on the type of the object. This operation again combines the animator control with the autonomous motion.

2.3 Facial Gestures

Face is one of the main streams of interaction among humans for representing intentions, thoughts and feelings; hence including facial expressions in the shared virtual environment is almost a requirement for efficient interaction. Although it is also possible to utilize a videoconferencing tool among the users in a separate window, it is more appropriate to display the facial gestures of the users in the face of their 3D virtual agent actors in 3D in order to give more natural virtual environment.

We include the facial interaction by texture mapping the image containing the user's face on the virtual actor's head. To obtain this, the subset of the image that contains the user's face is selected from the captured image and is sent to other users. To capture this subset of image, we apply the following method: initially the background image is stored without the user. Then, during the session, video stream images are analyzed, and the difference between the background image and the current image is used to determine the bounding box of the face in the image. This part of the image is compressed using the SGI Compression Library MVC1 compression algorithm. Finally, the image is sent to the other users after compression. There is a possibility to send uncompressed gray-scale images instead of using compression, which is useful if the used machines are not powerful enough to perform compression and decompression without a significant overhead. However, with all the machines we used this was not necessary. If this option is used, the compression can be turned on/off on the sending side, and the receiving side recognizes automatically the type of images coming.

At the receiving side, an additional service program is run continuously in addition to the VLNET program: it continuously accepts the next images for the users and puts to the shared memory. The VLNET program obtains the images from this shared memory for texture mapping. In this way, communication and simulation tasks are decoupled, decreasing the overhead by waiting for communication.

2.4 Fully Autonomous Actors

It is also possible to include additional virtual autonomous actors in the environment, which represent a service or a program, such as guiding in the navigation. As these virtual actors are not guided by the users, they should have sufficient behaviors to act autonomously to accomplish their tasks. This requires building behaviors for motion, as well as appropriate mechanisms for interaction.

Animation of autonomous actors is an active area of research [Thalmann94]. A typical behavioral animation system consists of three key components:

- the locomotor system,
- the perceptual system,
- the organism system.

The perceptual system should be improved through the synthetic vision [Renault-Thalmann90] for percepting the whole world, and appropriate mechanisms for interaction. Interaction with the virtual actors is also an active research area, and this should take into account multi-modal properties for communication.

2.5 Communication Architecture

The communication is based on a client/server model as illustrated in Figure 3. The server is designed as a lightweight applications making it possible to run the server continuously in background on any host without putting a noticeable strain on the CPU. This actually provides permanent virtual worlds to which the VLNET clients can connect, each being a kind of virtual meeting place.

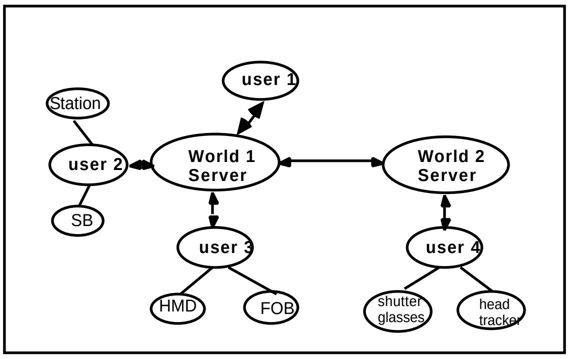

Fig. 3. CommunicationArchitecture of the VLNET System is based on a client/server model with link between the servers

When the client establishes connections with the server, the server first provides the scene description to the new client, including all the object files necessary to build and visualize the virtual environment. All the other clients are informed that a new user entered the virtual world. The user representation information (body description, face) is exchanged between all the users, passing through the server. This insures that each user can provide his own body and face description and thus be recognized by others.

Once this initial information exchange is finished, all information exchange is done through the server using uniformly sized packets which are not more than the Maximum Transfer Unit of the protocol being used. The content of each packet is interpreted according to its type - new transformation of an object, body skeleton angles, grouping/ungrouping information, entry/exit messages etc. The packet is a data structure comprising a header which contains the message type and the sender id, and the body which is a union of data structures -one for each message type. All geometrical information is sent in absolute, rather then incremental values, insuring the coherence of the shared virtual environment even if a packet is lost. When a user quits her VLNET session, the server cancels her from the client list and informs all other clients, thus insuring that this user disappears from the environment.

Links to other servers (i.e. other virtual worlds) can be established in a similar fashion as VRML or WWW links. These links can be attached to any object using a specialized motor function. When the user approaches such an object (it makes sense to make it look like a door), she is disconnected from the current server and connected to another one following the link. This gives the user the impression of "walking into another world". The user can take any objects with her when going to a different world. The linking mechanism, providing the possibility to carry objects through different worlds and allowing virtual actors to walk freely through the worlds, actually provides a hyper-world consisting of multiple servers scattered across the network

3 Implementation Issues

For the virtual environment to be realistic, our system should be fast enough for providing feedback; otherwise it is not comfortable to use. Therefore, we currently make use of the state-of-the-art technology discussed below to achieve currently utmost performance, but it is widely accepted that these platforms will be popular in a few years' time. There are three ways for speeding-up the system performance: display, communication, simulation.

For fast display, we make use of the IRIS Performer environment. Performer provides an easy-to-use environment to develop real-time graphics applications. It can extract the maximum performance of the graphics subsystem, and can utilize parallel processors in the workstation for graphics operations. Therefore, it is an appropriate platform to increase the performance of display part of the system. In addition, it provides efficient mechanisms for collision detection, and supports a variety of popular file types.

The network overhead can also have a significant effect, especially with increasing number of users. Therefore, it is important to provide low-latency high-throughput connections. The ATM network is one of the most promising solutions to the network problem, therefore we are experimenting our system over the ATM pilot network, provided to the Swiss Federal Institute of Technology and University of Geneva, by Swiss Telecom (Fig. 4.). The ATM technology, based on packet switching using fixed-length 53-byte cells, allows to utilize videoconferencing, video-on-demand, broadcast video. The quality of service is achieved on demand, and guarantees a constant performance. The network has full logical connectivity at the virtual path level and initially supports PDH 34 Mbit/s and SDH 155 Mbit/s links. The pilot provides point to point links in the first phase. Multipoint links are supposed to be added in the future, allowing more efficient multiple-user virtual environments. In addition, distributed virtual reality interface will allow to investigate new opportunities for different traffic and Quality of Service requirements.

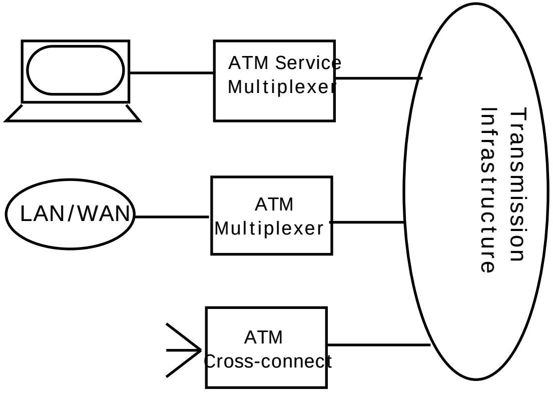

Fig. 4. Network Topology, ATM Overlay Network

In addition to the display and network parts, the simulation part should also be performed efficiently using appropriate mechanism. As discussed before, we make use of the HUMANOID system for modeling and real-time animation of virtual actors. The HUMANOID environment supports the following facilities:

- real-time manipulation of virtual actors on a standard graphics workstation,
- a way of accelerating the calculations using a dedicated parallel machine or by distributing the calculation load on several workstations
- a flexible design and management of multiple humanoid entities
- skin deformation of a human body, including the hands and the face

- a multi-layer facial animation module
- collision detection and correction between multiple humanoid entities
- several motion generators and their blending: keyframing, inverse kinematics, dynamics, walking and grasping

4 Connection to the Web

The 3D shared worlds described before are reserved to those who got the VLNET application. Without this specific program, it is impossible to access to the shared world, to follow his evolution : what is talking about, what are actors doing ?

By implementing a system to convert this enclosed world to a well known 3D language, we open the access of our shared world to the most part of the net population. This system is composed of an automatic save engine included in a classic VLNET client and a translator to obtain an image of the 3D world in a common language (Fig. 5.).

The save engine, permit to a client to take 3D snapshot of the scene. By 3D snapshot, we want to say that we collect all the 3D information of the scene at a given time. This 3D information is composed of the orientation and position of all movable objects but also of the position and the orientation of all clients connected at the same time.

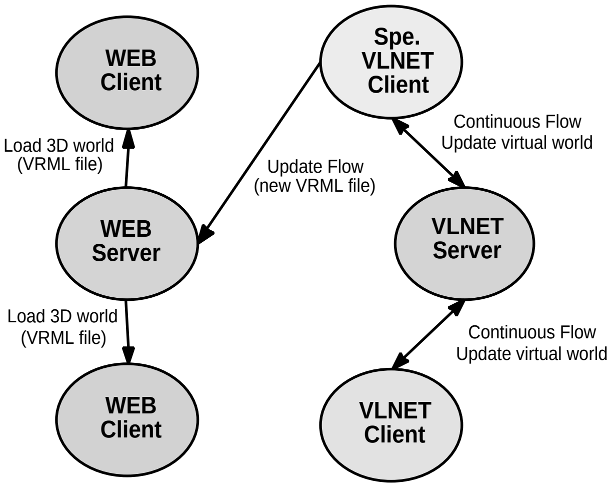
WEB Client
Spe. VLNET Client
Continuous Flow Update virtual world
Load 3D world (VRML file)
Update Flow (new VRML file)
WEB Server
VLNET Server
Load 3D world (VRML file)
Continuous Flow Update virtual world
WEB Client
VLNET Client

Fig. 5. 3D snapshot system.

The translator are then going to use this 3D information to generate a VRML file which will correspond exactly to the VLNET scene when the snapshot was made. As in VLNET, all connected clients are visualized by specific human surfaces, but a VLNET world, inside which actors are interacting, can be very huge and it can be very difficult for a snapshot user to find the representation of all VLNET clients. So, to obtain an easier way to walk through in our 3D snapshots by using a VRML browser, the translator define also some specific nodes to make directly accessible through a basic menu, the position of all actors by defining viewpoints : viewpoints are pre-defined positions for the camera and you can switch easily from one position to an other (Fig. 6.). By making available this final file through a classic WEB link, now, much more people can access to VLNET worlds to follow their evolution.

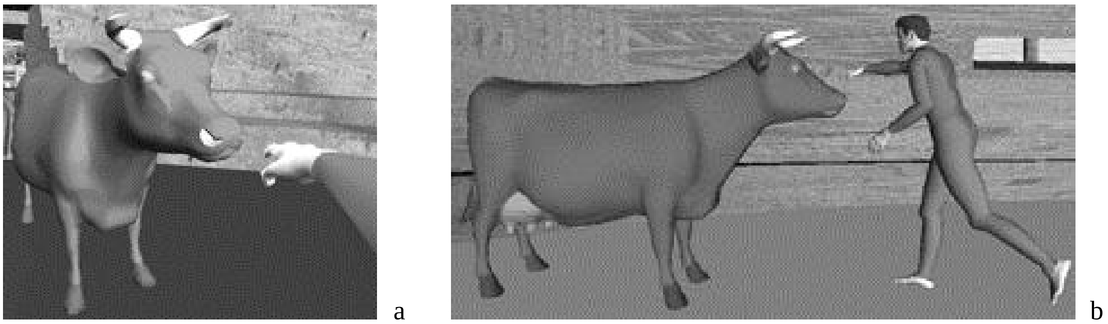

Fig. 6. The first image (a) was taken from a VLNET client.
The second image is from a 3D snapshot of the same scene.

4.1 VRML evolution and human representation in virtual worlds :

It is very important to introduce human like representation in those virtual worlds (VLNET, VRML) in order to reduce the distance that can exist between users and the world. It is more logical to interact with human like actors than implement some specific tools which only exist in those virtual worlds. A non-common user can then more easily understand what is going on and how he can interact with the virtual world.

The very first version of our snapshot system only convert the VLNET scene in VRML 1.0 format which is now obsolete compare to the features of the new version of VRML. But from this basic work, we can imagine how it is possible to improve our system in two principal ways :

- improve our snapshots.
- obtain more realistic human representation.

First of all, by using VRML 2.0 animation features, we can improve our snapshot by trying to animate the synthesis actors : for example, we can retrieve from VLNET, the direction in which each actors were moving when the 3D image of the shared world was taken and used this direction to perform a walk animation on few meters (according to the scale of the scene) or if a client was grasping an object when the snapshot was made, defining an animation in which we can see exactly the same thing as if we were in VLNET world. Only with this basic possibility of animation, we can give to the snapshots an impression of an evolving world in stead of a static world.

The second point is that with the system of prototyping of VRML 2.0, we can define a highly configurable human library. It means that we will be able to include different types of human in terms of size, thickness, level of details just by referencing the same VRML object but with different parameters. This library can include also the notion of hierarchy of the body and we can connect this VRML library very closely to the main existing human animation programs to be able to include more features like specific animations recorded from special devices like Flock of Birds from Ascension.

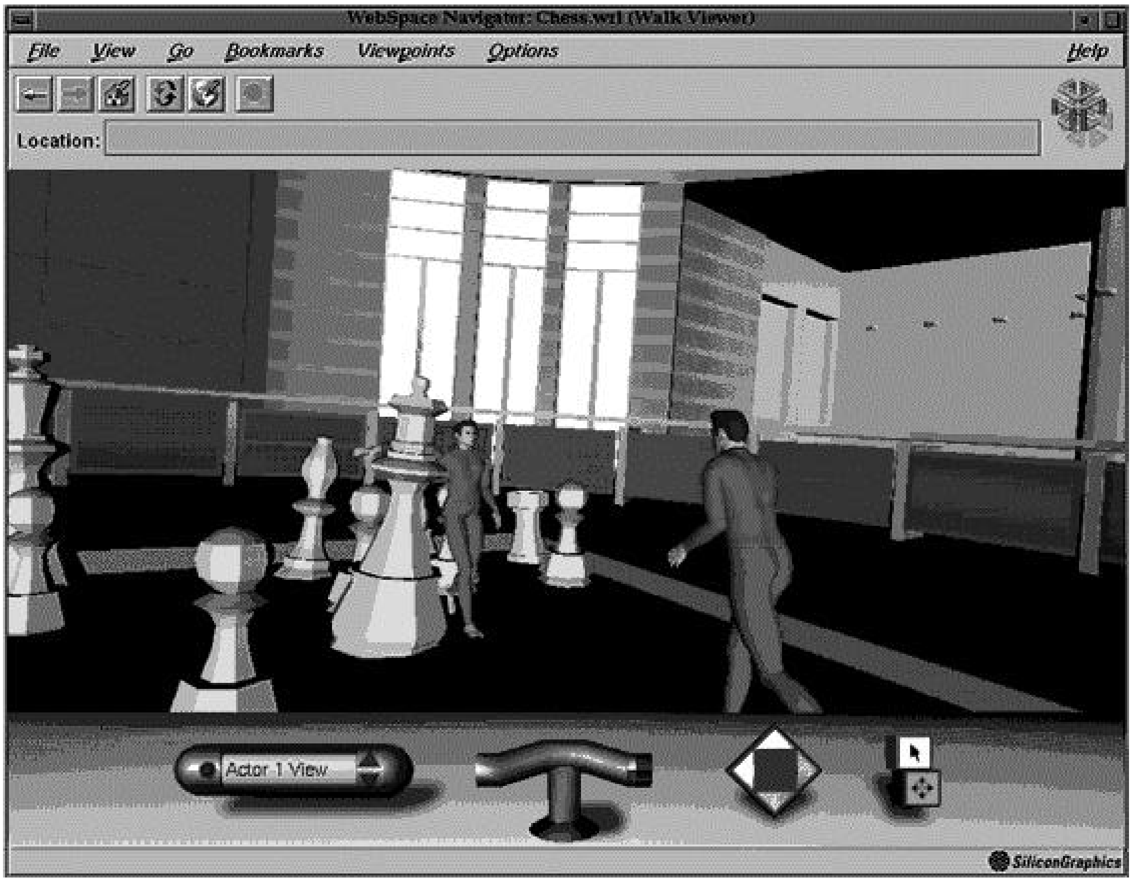

Fig. 7. A 3D snapshot of a VLNET chess scene with 2 actors visualized with*Webspace.*

5 Conclusions and Future Work

In this paper, we have presented a system which provides an efficient and visually effective way for human collaboration. Our system implementation should provide a satisfactory virtual environment for interactive cooperation.

Further improvements are to include physical models for interaction, more visual realism by incorporating already-developed body deformations module, natural language interface with the virtual actors, sound rendering. These new properties will require powerful processors but the accessability of VLNET shared world will be still available for all, through an improve system of 3D snapshots that will use new specifications proposed by VRML 2.0 *moving world*.

Acknowledgements

The research was partly supported by ESPRIT project HUMANOID (P 6079), Swiss National Foundation for Scientific Research, Silicon Graphics, l'Office Fédéral de l'Education et de la Science, and the Department of Economy of the State of Geneva. We would like to thank Riccardo Camiciottoli for his 3D Studio driver for Performer.

References

R. Boulic, T. Capin, Z. Huang, P. Kalra, B. Lintermann, N. Magnenat-Thalmann, L. Moccozet, T. Molet, I. Pandzic, K. Saar, A. Schmitt, J. Shen, D. Thalmann, The Humanoid Environment for Interactive Animation of Multiple Deformable Human Characters, in: *Proc. Eurographics '95*.

R. Boulic, N. Magnenat-Thalmann, D. Thalmann, A Global Human Walking Model with Real Time Kinematic Personification, in: *The Visual Computer*, Vol.6 (1990).

J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, M. Stone, Animated Conversation: Rule-Based Generation of Facial Expression Gesture and Spoken Interaction for Multiple Conversational Agents, in: *Proc. SIGGRAPH'94*.

L.E. Fahlen, O. Stahl, Distributed Virtual Realities as Vehicles for Collaboration, in: *Proc. Imagina '94*.

M.A. Gisi, C. Sacchi, Co-CAD: A Collaborative Mechanical CAD System, in: *Presence: Teleoperators and Virtual Environments*, Vol. 3, No. 4 (1994).

N. Magnenat-Thalmann, Tailoring Clothes for Virtual Actors, in: *Interacting with Virtual Environments* (Vince, MacDonald, 1994).

M.R. Macedonia, M.J. Zyda, D.R. Pratt, P.T. Barham, Zestwitz NPSNET: A Network Software Architecture for Large-Scale Virtual Environments, in: *Presence: Teleoperators and Virtual Environments*, Vol. 3, No. 4 (1994).

I.S. Pandzic, P. Kalra, N. Magnenat-Thalmann, D. Thalmann, Real-Time Facial Interaction, in: *Displays*, Vol 15, No 3 (1994).

J. Rohlf, J. Helman, IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics, in: *Proc. SIGGRAPH'94*.

O. Renault, N. Magnenat-Thalmann, D. Thalmann, A Vision-based Approach to Behavioral Animation, in: *The Journal of Visualization and Computer Animation*, Vol.1, No.1 (1990).

C. Rich, R.C. Waters, C. Strohecker, Y. Schabes, W.T. Freeman, M.C. Torrance, A.R. Golding, M. Roth, Demonstration of an Interactive Multimedia Environment, in: *IEEE Computer*, Vol. 27, No. 12 (1994).

Swiss Telecom 1993 "ATM Pilot Services and User Interfaces", in: *Swiss Telecom*.

S. Stansfield, A Distributed Virtual Reality Simulation System for Simulational Training, in: *Presence: Teleoperators and Virtual Environments*, Vol. 3, No. 4 (1994).

D. Thalmann, Automatic Control and Behavior of Virtual Actors, in: *Interacting with Virtual Environments*, (MacDonald, Vince, 1994).

D. Travis, T. Watson, M. Atyeo, Human Psychology in Virtual Environments, in: *Interacting with Virtual Environments*, (1994).

D. Zeltzer, M. Johnson, Virtual Actors and Virtual Environments, in: *Interacting with Virtual Environments*, (Vince, MacDonald, 1994).