Virtual Society:

Multi-user interactive shared space on WWW.

Kouichi Matsuda, Rodger Lea, Yasuaki Honda
Software Laboratory, Architecture Laboratories, Sony, Tokyo, Japan
Tel: +81 3 5448 5437 email: matsuda@arch.sony.co.jp

Abstract

The Virtual Society project is a research initiative to investigate the future electronic society. Our vision for such a society is a based on the notion of shared multi-user 3D virtual worlds where people can meet, interact and work. People will also be able to enjoy multiple lives in the world and take on any of a range of personalities and roles. We have developed the CommunityPlace system to realize an initial implementation of our vision for the Virtual Society. It is targeted at the WWW and uses the Virtual Reality Modeling Language (VRML). We believe that such a system will become the next generation user interface over the Internet. In this paper, we discuss the architecture and implementation of the CommunityPlace system, Internet-related technology such as VRML, the latest VRML-related activities and our future plans.

Keywords: Distributed Virtual Environment Internet VRML

1. Introduction

The term "Cyberspace" although popularized by the media exists today only in science fiction novels. However, the recent evolution of computer technologies and network technologies provides the enabling infrastructure for cyberspace. The goal of Virtual Society project is to provide the cyberspace, or a massive shared multi-user 3D virtual world on WWW. People can access the world from the anywhere in the Internet, from homes, schools and offices. We wish to create worlds where many people can meet, interact and work together and investigate how the future electronic society will evolve. Also people will be able to enjoy multiple lives in the world. The investigation such people's activities in the world will become a cyberspace cultural anthropology in the future. We also believe that such a world will become the next generation of the user interface over the Internet.

This project started in the middle of 1994 at Sony Computer Science Lab. The first goal was to build a support infrastructure that allowed many people to participate in a shared, interactive 3D world. Such interaction will include the ability to see each other, talk to each other, visit locales with each other and share the events happen in the shared world. The CommunityPlace system has been developed in the project as the support infrastructure for an initial version of Virtual Society. The first release (based on VRML1.0) was at the end of 1995; subsequently we have migrated to VRML2.0 and made a version of the system available since early 1996.

2. CommunityPlace

CommunityPlace (previously known as CyberPassage) is the name of a suite of software developed by Sony's Architecture Laboratories to realize the Virtual Society. It is targeted at hom

PC's, dial up lines and the Internet. It consists of four software components; a 3D browser known as CommunityPlace Browser, a 3D scene and behavior construction tool, known as CommunityPlace Conductor, a multi-user server known as CommunityPlace Bureau and an application environment known as AO. To construct virtual worlds on WWW, we chose the VRML as a 3D file format. We discuss the VRML in the next section.

CommunityPlace Browser is a PC based browser to display and navigate through virtual 3D worlds. It supports VRML1.0 and VRML2.0 as a 3D file format and Java as a scripting language to describe behaviors. In addition, it communicates with the CommunityPlace Bureau to support shared 3D worlds by using a client server protocol Virtual Society Client Protocol (VSCP) to send and receive updates about changes to scene geometry. In the shared worlds, The CommunityPlace browser supports on-line text chat and audio chat. The CommunityPlace browser can work as both a helper application and plug-in for Netscape Navigator. Currently Windows 95/NT versions are freely available for download from our web site. Figure 2.1 shows the CommunityPlace Browser connecting to a multi-user world.

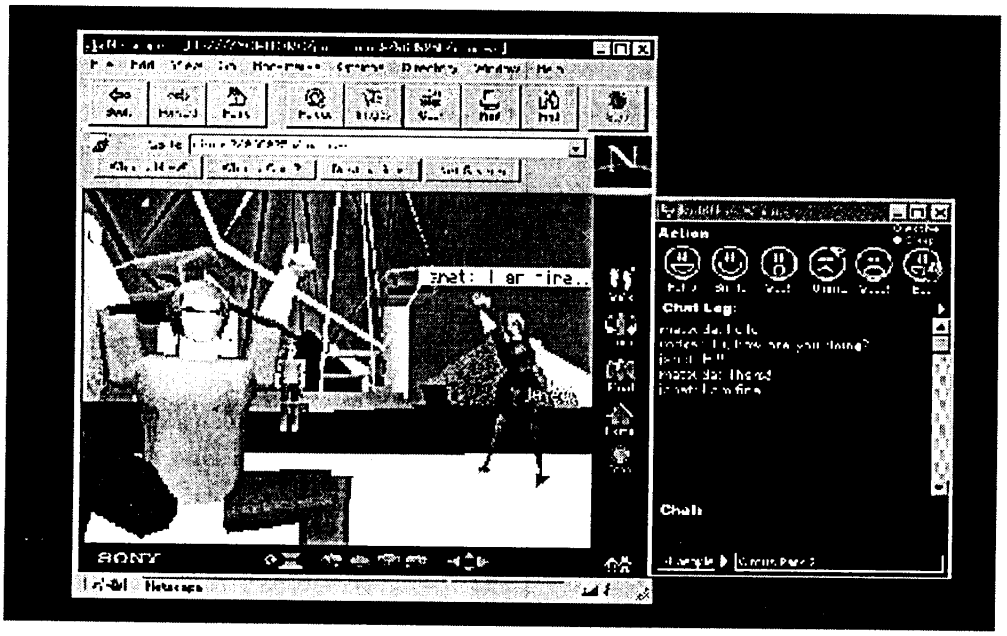

Figure 2.1 CommunityPlace Browser connecting to a multi-user world

CommunityPlace Conductor is a PC based authoring tool to create virtual 3D worlds. It is a kind of integration tool and does not support 3D modeling features. Rather it is used to compose scenes from 3D objects created in modeling tools to which can be added multimedia data (sound and video), textures and behaviors. This approach allows the creation of interactive, multi-media 3D scenes or worlds. Community Place conductor supports VRML1.0 and VRML2.0 as its 3D file format and uses Java as a scripting language. Currently Windows 95/NT versions are available. Figure 2.2 shows the CommunityPlace conductor.

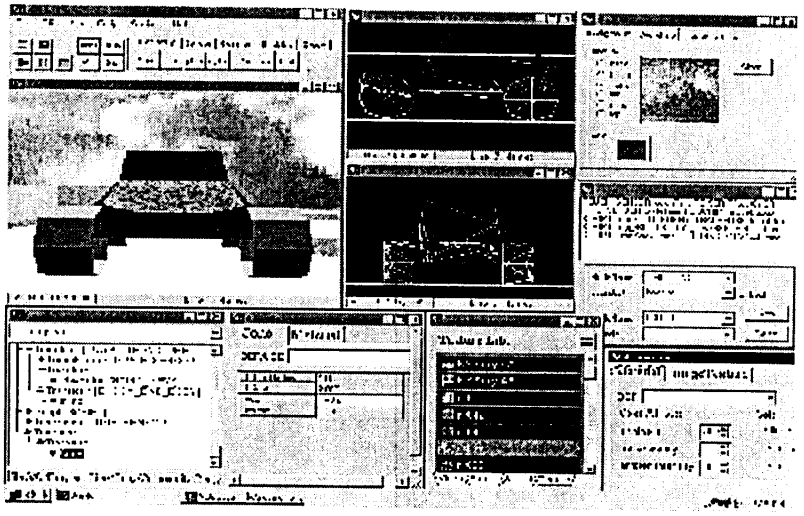

Figure 2.2 CommunityPlace conductor

CommunityPlace Bureau is a PC and UNIX based server system to share virtual 3D world among users. It receives 3D geometric changes and messages from user's browser or external applications for the shared 3D worlds and distributes these changes and messages to other users so that they can visualize the changes. Currently Windows 95/NT and various flavors of UNIX versions are available. The UNIX version can support up to 600 users simultaneously for one world.

The application environment (AO) is a separate component that provides an API suitable for building 3D applications. The AO connects to the server and registers any application objects created via the AO API. These are then dynamically added to the shared 3D scene.

Both the browser and conductor can run on ordinary home PCs and do not require any special hardware like a data glove, HMD(Head Mounted Display) or 3D rendering hardware. So user can navigate a 3D world with a mouse and the 3D world is rendered in real-time to the computer display by 3D rendering software engine. We refer to this type of Virtual Reality experience as fish tank virtual reality.

3. CommunityPlace system architecture

It is possible to build multi-user 3D applications by using the CommunityPlace system. In this section, we discuss how the CommunityPlace system supports such application development.

In addition to the three components described above, the CommunityPlace system assumes a WWW server and HTML browser. The WWW server manages VRML files which the HTML browser retrieves over the Internet. The VRML files contain a description of the virtual 3D world and multi-user applications. It also has a IP address and port number of the computer where CommunityPlace Bureau is running.

When the HTML browser accesses the VRML file, it is passed to the CommunityPlace browser. Then the CommunityPlace browser analyzes the file and visualizes it. If the VRML file is designed as a multi-user content, the CommunityPlace browser tries to connect to the CommunityPlace Bureau specified in the VRML file.

After connecting to the CommunityPlace Bureau, the rest of the communication between th

Bureau and the browser is performed using VSCP and only uses HTTP to access files from WWW servers. The basic function of VSCP is to notify the server about any changes made by the users via their browser and to be notified of any changes made by other browser. This allows any browser which access the same VRML file to connect to the same CommunityPlace Bureau and so share the VRML scene. Using this approach, any VRML scene can be shared.

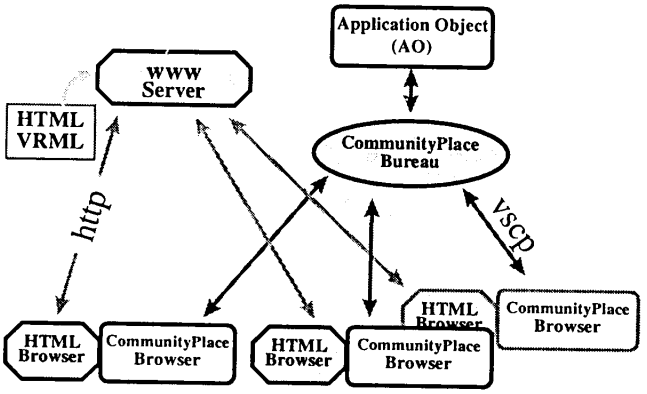

Figure 3.1 CommunityPlace system architecture

In the shared 3D world, each user accessing to the same world is represented as an avatar, or a 3D representation of the user, in the world. Thus, each user's behaviors, e.g. navigating through the 3D scene is reflect by his/her avatar in all other browser sharing the scene.

When a user navigates around the 3D world by moving and rotating his/her avatar, this movement and rotation information is sent from the CommunityPlace browser to the connected CommunityPlace Bureau using the VSCP protocol. Then the CommunityPlace Bureau propagates the information to all other CommunityPlace browser connected to the Bureau. When a CommunityPlace browser receives the information from the server, it updates the corresponding user's avatar according to reflect the changes made by the source browser.

Multi-user applications in the shared world are described using the multi-user application programming model designed for CommunityPlace system. We will discuss the model later.

4. VRML

The Virtual Reality Modeling Language (VRML) is 3D file format designed to create virtual 3D worlds for delivery over the Internet. The language used for text documents in the Internet is Hyper-Text Markup Language(HTML). VRML can be thought as a 3D version of HTML and is being standardized by the International Standards Organization (ISO). The latest version of the standard is know as VRML2.0 and has been adopted by most major VRML browser and tool vendors.

4.1 VRML1.0

VRML1.0 is the first version of VRML and it was released in May 1995. VRML1.0 is a simple graphics language based on the OpenInventor's ASCII file format of SGI. VRML1.0 described 3D worlds from a series of transformation node, i.e. spatial position marker. Each node can contain sub-nodes forming tree (called "scene graph"). VRML1.0 provides a set of geometry nodes (cone, sphere, etc.), property nodes (color, texture, etc.), grouping nodes and special nodes (camera, light, etc.). It also supports WWW-oriented extensions to enable authors to specify texture files as a URL, add a link from 3D objects to any URL. This allows 3D scene authors to integrate the scene with the WWW.

Figure 4.1 shows a red sphere in VRML1.0.

```
#VRML V1.0 ascii
Separator{
    Material { diffuseColor 1 0 0 } # red (RGB)
    Sphere { radius 10 }
}
```

Figure 4.1 A red sphere (VRML1.0)

A VRML browser acts as an interpreter of the ASCII text files written in VRML and renders in real time a visual form of the file. As the user navigates the virtual space with the browser interface, the browser regenerates the scene in real-time from the user's viewpoint.

VRML1.0 is an adequate language for describing static world. But it does not support our requirements to realize the Virtual Society for creating and sharing interactive world. We extended VRML1.0 to solve these problems. The extension is known as E-VRML (Enhanced VRML). E-VRML supports several extension nodes that support sound, video and a mechanism to associate a programming language with a 3D scene object. By using the mechanism, it is possible to add behaviors to the 3D objects.

Subsequently these extensions were combined into a joint proposal with SGI and Worldmaker called "Moving Worlds", are submitted to the VRML Architecture Group (VAG) as a proposal for VRML2.0. This proposal was accepted as the basis for VRML2.0 and after revision, was officially released in August 1996.

4.2 VRML2.0

VRML2.0[1] is also a language for describing 3D scenes like VRML1.0 but it can also be used to create interactive 3D world with sound, video and behaviors.

VRML2.0 retains the node structure of VRML1.0 and makes up a scene from a series of transformation nodes. In addition, it supports multimedia nodes which enable ambient sound or spot sound and video. It also adds a mechanism to associate a programming language fragment with 3D scene objects. The mechanism consists of four sub-mechanisms; sensor, event, routing and script.

The sensor is the mechanism to detect external events and convert them to the internal VRML events. For example, VRML2.0 supports a TouchSensor to detect user's mouse operation and generate internal VRML events. The event is the mechanism that passes information between nodes and goes so by using a routing mechanism. The target for routing events may be graphical nodes, or more typically, script nodes. When the events are routed to a script node, the associated program is executed and passed an events send to the script node. The program can then generate new events

that are passed back to the script node. These events may then be routed to other nodes and change the scene graph (for example, changing color of 3D objects or moving them).

Currently either Java or VRMLscript is available as a scripting language(SGI's CosmoPlayer browser supports VRMLscript, Sony CommunityPlace browser and DimensionX's Liquid Reality browser supports Java).

4.3 VRML2.0 Example

Figure 4.2 and Figure 4.3 show the example which changes the color of sphere from red to blue by using Java as a script language. Figure 4.2 describes a world and Figure 4.3 shows the script bound to the world.

```
 1 #VRML V2.0 utf8
 2 Transform {
 3     children [
 4         DEF TS TouchSensor {}
 5         Shape {
 6             appearance Appearance {
 7                 material DEF SphereColor Material { diffuseColor 1 0 0 }
 8             }
 9             geometry Sphere {}
10         }
11     ]
12 }
13 # Script node
14 DEF ChangeColor Script {
15     url "ChangeColor.class"
16     eventIn SFBool clicked
17     eventOut SFColor newColor
18 }
19 # Routing
20 ROUTE TS.isActive TO ChangeColor.clicked
21 ROUTE ChangeColor.newColor TO SphereColor.set_diffuseColor
```

Figure 4.2 A red sphere (VRML2.0)

In this example, when the user clicks the sphere, the TouchSensor (line 4) detects the event and then generates an VRML event from the isActive field of the TouchSensor. This event is routed to clicked field of Script node (line 16) according to the specified routing (line 20). Then it is passed to ChangeColor.class(Figure 4.3). After completing some calculation in the program, it generates an event (color) on newColor field of the Script node (line 17). It is routed to diffuseColor field (line 21). Consequently, the sphere's diffuse color is changed.

```
 1 // ChangeColor.java
 2 import vrml.*;
 3 import vrml.field.*;
 4 import vrml.node.*;
 5
 6 public class ChangeColor extends Script{
 7     private boolean on = false;      // status of on/off
 8     float red[] = { 1, 0, 0 };       // RGB(Red)
 9     float blue[] = { 0, 0, 1 };      // RGB(Blue)
10     private SFColor newColor;        // reference to "newColor" field of
11
12     public void initialize(){
13         newColor = (SFColor) getEventOut("newColor");
```

```
14      }
15
16      public void processEvent(Event e) {
17          ConstSFBool v = (ConstSFBool)e.getValue();
18
19          if(v.getValue()){
20              if (on) {
21                  newColor.setValue(red); // set red to 'newColor'
22              } else {
23                  newColor.setValue(blue); // set blue to 'newColor'
24              }
25              on = !on;
26          }
27      }
28  }
```

Figure 4.3 Script to change a color of sphere

When the event is passed to ChangeColor.class, processEvent() method (line 16) is called with the event as an argument. In the method, a new color is set to the reference to newColor field. The reference is gotten using getEventOut() in initialize() method (line 13) that is called before any event is generated.

In addition to the VRML2.0's behavior mechanism, the CommunityPlace system supports Java classes to enable to build multi-user applications. Its fundamental function is to send messages to the CommunityPlace Bureau using VSCP. The messages from other browsers appear on Script node's eventIn field and it is also passed to the processEvent() method.

5. CommunityPlace Bureau

The CommunityPlace Bureau acts as a position tracker and message forwarder in CommunityPlace system. Its primary role to receive the user's position and the information made by the browser and them distribute them to the other browsers. In the current system, due to performance and scalability consideration, the Bureau also has the role to restrict the distribution of information to those browsers that need to know. It uses an Aura [6] algorithms to decide which other browsers need to be sent the information. An Aura is the area around a user that is deemed to be interesting, anything outside the aura is not considered interesting. So the Bureau does not send the information of the browsers outside the aura to the browser which has the aura. This spatial technique allows us to scale the system. Figure 5.1 shows how Aura works. In this case, user A and user B can see each other but they can not see user C and user C can not see them.

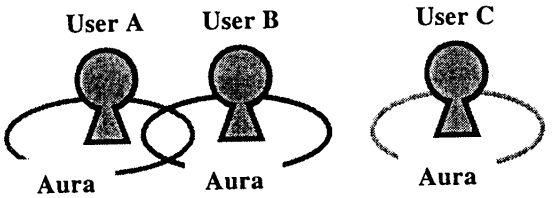

Figure 5.1 Aura

In addition to the aura, the CommunityPlace system supports World Location Server (WLS) to enable to switch multiple Bureaus dynamically. WLS enables to divide one shared world into several

sections geometrically and assign the Bureaus to each section. Figure 5.2 shows how WLS works. A user(browser) is in the area1 where the Bureau1 manages and then enters into the area2 where the Bureau2 manages. Bureau1 sends a leaving message to the browser (1) and the browser asks the WLS about the next Bureau to be connected (2) and it sends the information of the Bureau2 (3). Then the browser connects to it (4).

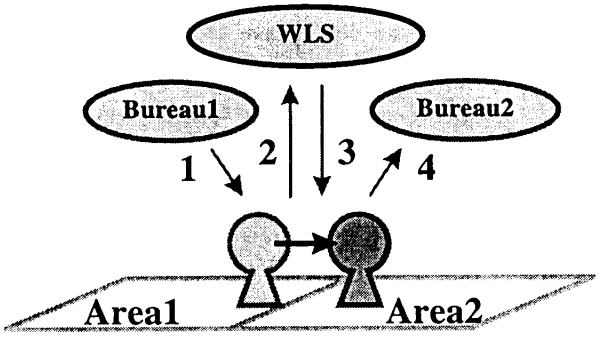

Figure 5.2 World Location Server

6. Multi-user application programming model

The CommunityPlace system provides two types of application programming model for building multi-user applications. The first one is called the Simple Shared Script (SSS) model and the second one is called the Application object (AO) model.

6.1 Simple shared script

The SSS is a simple mechanism designed for small shared application in the 3D world. It uses a replicated script model. So each browser uses the same script and executes it locally. The script can send a message to all other browser sharing the world using VSCP via CommunityPlace Bureau.

Figure 6.1 shows how SSS model works. When a user selects a 3D object(1), a local script associated with the object is executed (2). Then the script converts the event into a message and sends it to CommunityPlace Bureau(3). The Bureau distributes the message to all other browser (4). The browser convert the message into an event that causes the local script execution (5).

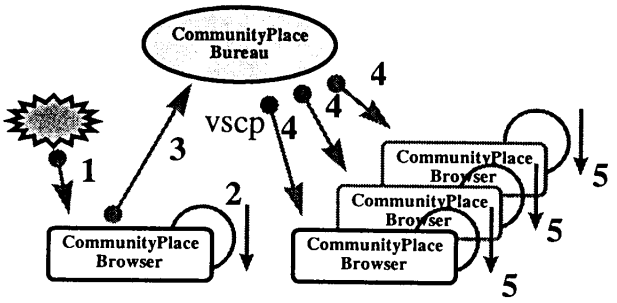

Figure 6.1 Simple Shared Script model

The drawback of the SSS model are based on ownership and persistence. Since all scripts are equal, they need to communicate among themselves to resolve ownership and persistence. To solve these problems, we introduced a mechanism to assign master ownership to one browser among the browsers connecting to the same shared world. It can receive messages from other browsers and send messages to them. We tend to use the SSS for simple shared applications that do not need sophisticated synchronization or persistency requirements.

6.2 Application object

The application objects are the entities which are responsible for handling application specific behavior of 3D objects in the 3D world. It allows application creators to create 3D objects and inject them in the existing shared scene dynamically. It also supports to delete the objects dynamically. The AO sits out of the CommunityPlace Bureau and communicate with it using Virtual Society Application Protocol(VSAP) protocol to register the AO and send requests and receive a message to/from the Bureau.

Figure 6.2 shows how AO model works. When AO is attached to the Bureau, it requests the Bureau to create and inject 3D objects (1). The Bureau sends the request to all browser and then the browser do that (2).

When a user selects a 3D object(3), a local script associated with the object is executed (4). The script sends a message to the CommunityPlace Bureau (5). The Bureau sends the message to the AO managing the selected object (6). The AO performs internal processing and then typically sends back a message (7) via the Bureau to all other browser (8).

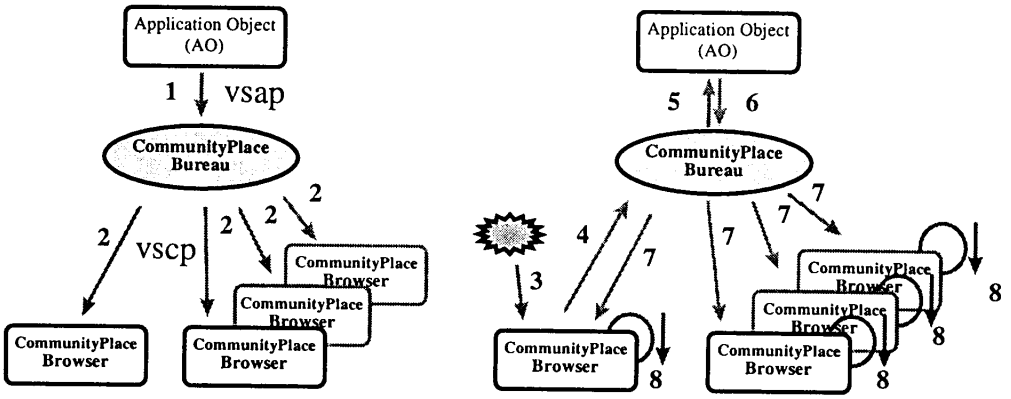

Figure 6.2 Application object model

7. Current status

The CommunityPlace system has been in public beta release since December 1995. The downloadable browser, conductor and server are available. The system is fully functional and supports distributed shared world, multi-user chat feature (both text and audio) and shared behaviors. It is based on the WWW, VRML2.0 and Java standard. The product version has been available since September 1996.

We have tested larger servers at two public sites, one is in Japan and one in the USA since February 1996. Both sites are freely accessable from anywhere on the Internet to connect to and share a world. We have made available a set of sample shared worlds. Figure 7.1 shows one of the sample

shared worlds called "Circus Park" and the shared behavior. In the figure, we can see two browsers showing the same scene from two different viewpoints. In the center is a sea lion whose behavior is to flip the ball up on his nose and put it down again. This behavior is activated by user selection and shared. So the all browsers within the sea lion's aura can see that the behavior is happening simultaneously.

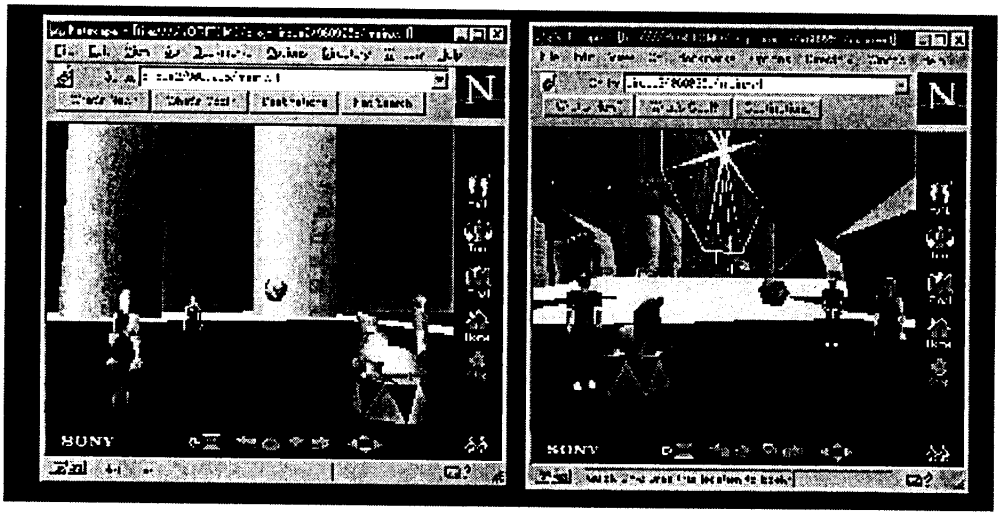

Figure 7.1 Shared behavior

Our initial experiences are encouraging, we have fostered a small but active community of regular visitors at the public site in Japan. Users have had no difficulty driving the system and making use of the interaction mechanisms we have provided.

We have limited experience of collaboration within our 3D framework having built only a simple shared whiteboard. However, we are now developing more sophisticated shared applications and hope to deploy them within several months.

8. Related work

Recent work targeting the WWW and using full 3D shared spaces has mainly been confined to the VRML community. Within that community there are several projects of note.

The **Cybergate** system from Blacksun Inc.[7] has built and experimented with shared 3D spaces similar to the CP project. However, CyberGate is based on CosmoPlayer and Live3D and so supports only static scenes. **Moondo**[8] from Intel is a similar system to CyberGate in that it currently supports only static scenes. However, Moondo has experimented with a shared object model as a basis for shared consistent objects. In addition, Moondo has added support for audio chat.

The **Pueblo** project from Chaco Communications [9] has evolved out of earlier work on social MUDs (Multi-User Dungeon). Recently it has augmented the MUD server with VRML support and provided a VRML1.0 browser that allows MUD authors to build 3D scenes. This approach allows world builders access to the rich mechanism of the MUD database, but again only supports static scenes.

An interesting experiment in 3D spaces that supports audio chat is the**Traveler** system from

OnLive![10] Again, Traveler is VRML1.0 based, and so static, but the primary focus of the group has been on audio interaction. To support this, they have built a low-bandwidth audio codec suitable for the Internet. In addition, they have augmented user avatars with facial motion, and in particular lip synch, that is driven from the audio stream. This technique offers a computationally cheap, but rich and compelling interaction mechanism.

9. The latest VRML-related activities.

After releasing VRML2.0, there are three major VRML-related activities; binary file format, external authoring interface and standardization for realizing a multi-user system. These activities are being discussed in www-vrml mailing list.

Binary format is a compressed file format for VRML2.0 files which enable to transmit them efficiently [11]. Currently VAG called for proposals for the binary format. IBM, Apple Computer and ParaGraph submitted their proposal. Their proposal consists of geometry compression and translation of VRML2.0 ASCII syntax into binary tokens. IBM will provide the reference implementation of the compression/decompression algorithms. ParaGraph will develop the reference implementation of the ascii/binary parser.

External Authoring Interface is the standard of application programming interface to enable user to build authoring tool externally on the VRML browser. Once the authoring tool is built with the EAI, the tool becomes available for other EAI-enabled VRML browser. Two proposals, External Authoring Interface (SGI)[12] and External API (DimensionX)[13] were submitted to VAG. Their APIs are based on JAVA. They are being putted to the vote.

Living Worlds[14] is one of the proposals to realize multi-user applications by using VRML2.0. It was proposed by Sony, Black Sun Interactive and Paragraph. This proposal focuses on node extensions to support the applications but does not focus on protocol, network or server systems. These components are left for implementers to enable to develop their own components. The proposal guarantees that "Living Worlds" compliant contents will become available on all "Living Worlds" compliant browser. The first draft was opened to public in October 1996.

10 Conclusion and future plans

Our current system is sufficient for medium scaled shared social spaces. We are pursuing several developments to allow our platform to evolve into a general purpose shared 3D platform supporting the Virtual Society concept.

Our first area of concern is scaling. Our current server supports up to 700 users. We have a goal to support several thousand users for mass events such as shared concerts etc. To support this we are currently developing a distributed multi-user server. The distributed server uses multicast communication to allow replication of server data.

A second area of development is tools and applications to enrich the shared worlds. Our first development is support for audio chat in addition to textual chat. Subsequently we will add support for streaming audio and video into shared worlds. In parallel we are developing a general purpose application environment with a rich set of application API's to allow anybody to author shared 3D scenes.

Lastly we are exploring techniques to foster on-line communities, these include:

- Supporting a social infrastructure
- Supporting social rules
- User interface to interact with other people
- Concept of ownership in the shared virtual world

Our goal is to provide a rich 3D collaborative experience through a combination of base level technology and higher level tools. This will encourage third parties to use and develop new technology for our platform.

The Community Place system is freely downloadable from two web sites:

http://vs.sony.co.jp

http://www.spiw.com/vs

1. Acknowledgments

We are indebted to our colleagues in the Sony Computer Science Lab. and Sony's architecture Lab. for their help in the definition of this project. We also wish to thank our colleagues at the Swedish Institute of Computer Science who have contributed indirectly to the CP design as part of our joint research project, Wide area virtual environments (WAVE). Lastly, our thanks, as always, go to Akikazu Takeuchi, Mario Tokoro and Toshi Doi for their continuing support.

References

1 The VRML2.0 specification. Currently available as http://vag.vrml.org/VRML2.0/FINAL

2 Honda, Y., Matsuda, K, Rekimoto, J and Lea, R. Virtual society. Procs. of VRML'95, San Diego. USA. Dec. ACM press 1995 pp. 109-116, Order no. 434953 Available at: http://www.csl.sony.co.jp/project/VS/VRML95.ps.Z

3 Lea, R., Raverdy, P.G, Honda. Y, and Matsuda, K. Issues in the design of a large scale VE. Sony Computer Science Lab Tech Report XX.XX.95 available from http://www.csl.sony.co.jp/

4 Lea, R., Raverdy, P.G, Honda. Y, and Matsuda, K. (Virtual Society: Collaboration in 3D spaces on the Internet. available from http://www.csl.sony.co.jp/

5 Carlsson, C. and Hagsand, O. DIVE - A platform for multi user virtual environments. Computer and Graphics Vol. 17. No. 6 1993 pp. 663-669

6 Benford.S, and Fahlen, L. A spatial model of interaction in large virtual environments. September 1993, In proceedings of G. DeMichelis et al (Eds.) Third Europian Conference on Computer Supported Cooperative Work (pp. 109-124), Kluwer Academic Publishers.

7 Blacksun Inc. home page. Currently available as http://www.blacksun.com/

8 Intel Moondo home page. Currently available as http://www.intel.com/iaweb/moondo/index.htm

9 Chaco Communications home page. Currently available as http://www.chaco.com/DD>

10 OnLive! home page. Currently available as http://www.onlive.com/index.html

11 Binary format proposal. Currently available as http://www.austin.ibm.com/vrml/binary

12 SGI's external authoring interface proposal. Currently available as http://vrml.sgi.com/moving-worlds/spec/ExternalInterface.html

13 Dimension X's external interface proposal. Currently available as http://www.dimensionx.com/people/wing/extapi.html
14 "Living Worlds" proposal. Currently available as http://www.livingworlds.com/draft_1/index.htm