The Petri Net Model for the Collaborative Virtual Environment on the Web

Jiung-Yao Huang and Lawrence Y. Deng

Multimedia & Virtual Reality Lab.
Department of Computer Science and Information Engineering
Tamkang University,
Tamsui, Taipei, Taiwan 251
E-mail address: jhuang@mail.tku.edu.tw

Abstract

This paper presents a Petri Net model to analyze the workflow of a web-based multiple participants virtual environment. The presented approach not only can conspicuously help the developer to comprehend the interaction relationship between the client-server virtual environments but also to easily construct a shared virtual world. Based on the presented Petri Net model, we propose an architecture for the collaborative virtual environment and implement a multiple user 3D web browsing system, called the SharedWeb system. Problems of providing the multi-user interaction on the Web and the solutions proposed by the Petri Net model are fully elaborated here. Some experimental results along with two demonstrated virtual world are also presented.

Key Words : World Wide Web, Common Gateway Interface, Distributive Interactive Simulation, Dead Reckoning, Petri Net and Virtual Reality.

1. Introduction

Virtual reality is a technique to immerse the user into a synthetic world created by the computer or into a remote environment electronically transmitted to the computer for the purpose of training, simulation, prototyping, design, or data acquisition. The multi-user virtual environment is a computer-generated synthetic world on the network where users can navigate and interact with each other. Due to the rapid evolvement of Internet technology, researches on the multi-user virtual environment have become popular in the computer graphics community over the past years.

The World Wide Web (WWW) is a project to bring a global information universe to the existence.[1] The Virtual Reality Modeling Language(VRML) is an attempt to deliver a virtual reality world into the WWW environment.[2] The VRML standard enables the user to design 3D multimedia and shared virtual worlds, which are interconnected together via hyperlinks, on the Internet. After the introduction of the VRML language, many VRML-supported browsers were developed, including the WebSpace from SGI, WebOOGL from the Geometry Center, AmberGL from DIVE Lab., WorldView for Intervista Software, Inc. With these browsers, you can view and walk through a virtual world described by the retrieved VRML file.

The original goal of designing the VRML standard is to allow multiple participants to interact over the Web environment. However, even for version 2.0 that was published in August 1996, this design goal was not fulfilled. In addition, none of the above browsers provide the multiple-user interaction capability. The development of VRML has nevertheless spurred broad research on distributed multi-user virtual reality systems.

The following sections begin with a survey of related works on the multiple user virtual reality system, and followed with the Petri Net model of a seamlessly integrated multiple user virtual reality

system. The implementation and experiments of the SharedWeb system based upon the Petri Net model will be discussed, along with some experimental results that show its potential applications including the Virtual Campus and Virtual Office.

2. Related Researches

The researches on the multiple user virtual environments can be classified into two categories: network-based and web-based virtual reality systems. The first category of the multiple user virtual reality system does not necessarily comply with the architecture of the Web environment. This type of virtual reality system generally focuses on providing a large-scale multi-user virtual environment with every available network technique. On the other hand, the web-based multiple user virtual reality system emphasizes on providing collaborative virtual environments on top of the existent Web environment. Although this paper focuses on modeling a multiple user virtual reality system on the existent Web environment, the techniques from the network-based multiple user system also provide partial solutions to our work. Hence, the techniques used by the network-based virtual reality system will be discussed first.

2.1 Network-based Collaborative Virtual Reality Systems

There are two issues to be solved when a multi-user virtual environment over the Internet is built. One is the spatial consistency problem and the other is the time coherency problem. Due to the intrinsic network latency, time coherency is almost an insolvable problem for the multiple participants collaboration over the network. Hence, most of the researches of the networked virtual environment have focused on the spatial consistency problem. Various methods have been proposed to maintain the consistency of the synthetic world among distributed players. These methods include communication protocols to log in an existent virtual world and to share the status of objects among distributed participants.

In addition, these communication protocols are tightly related to the database models used by the networked virtual reality systems. According to the distribution of the virtual world database, Macedonia and Zyda [3] classified the networked virtual environments into four types:

- **Replicated homogeneous world database**. SIMNET [4] and NPSNET [5] are two delegations of the replicated homogeneous world model. The NPSNET system uses the DIS (Distributive Interactive Simulation) standard [6], which is descended from the SIMNET, to communicate the entity status changes among distributed players. The DIS is an IEEE standard for multi-user interaction over the Internet, which will be discussed in the following subsection.
- **Shared and centralized database**. RING [7] and VLNET [8] are two systems that employ the shared and centralized database model. VLNET focuses on providing realistic virtual human characters inside the virtual world. To achieve such a goal, different protocols were designed for the object behavior, navigation, body posture, and facial expression.
- **Shared and distributed database with peer-to-peer updates**. Similar to SIMNET, DIVE [9] system has fully duplicated homogeneous and distributed databases. However, the entire databases are dynamic and a reliable multicast protocol is used to actively replicate new objects.
- **Shared, distributed and client-server databases**. In order to scale up the number of participants without compensating the shared object consistency, BRICKNET [10] and MASSIVE [11] systems employ shared, distributed and client-server databases to achieve the goal. In addition, BRICKNET took one step further to allow objects' dynamic behaviors to be shared among distributed players. MASSIVE uses a spatial model for database partition among clients. These features require complex protocols to ensure strict harmony among distributed sites.

Although the above systems were proven to provide effective distributed virtual worlds, none of them worked on integrating the multi-user system with the Web environment. With the emergence of the VRML standard that brings the virtual world into the Web environment, researches on supporting the multi-user interaction on the Web environment have become a popular topic. Hence, a survey of the previous works on providing multiple user interactions on the Web environment will be discussed next.

2.2 Web-based Multiple Participant Virtual Reality Systems

The Web environment is a client/server architecture, and it is widely used to share information over the Internet. Due to the intrinsic

characteristic of the HTTP that is used by the Web architecture, the traditional Web system is a "one way street". In other words, it is impossible to achieve the interactivity among users under the existent Web architecture. Hence, mechanisms and protocols have to be designed to support the multiple user interactions on the Web environment. The VRML standard is an attempt to deliver the virtual world on the Web, and its original goal is to support the interaction of multiple participants. However, even the VRML version 2.0 [2], which was released in August 1996, failed to define the standard to meet the goal. Hence, several researches on the web-based collaborative systems have their own proprietary methods to support the multiple user interaction [12, 13].

The Virtual Society (VS) is a research project of Sony Computer Science Laboratory Inc. [13]. This project is an attempt to define a global architecture and a set of protocols to realize a multi-user interactive 3D environment in a WWW setting. The VS project was originally based on the DIVE platform from the Swedish Institute of Computer Science [9], and a so-called Community Place browsing system [14] was designed for this project. Different from the DIVE system, the Community Place browsing system is a VRML-standard virtual reality system on the Web environment.

In order to support large-scale shared 3D spaces using the VRML, the Virtual Society proposed a so-called Virtual Society Server Client Protocol (VSCP) to extend the function of the existing VRML standard. In addition, a scripting architecture is also designed for different levels of consistency for the shared objects. However, the Virtual Society architecture is not seamlessly integrated with the Web environment. For example, the VS client cannot enter a virtual world from a Web server form which the scene file was downloaded. Instead, after a scene file is successfully downloaded, the VS client must send a URL to that Web server to query the information of the VS server. The Web server then returns an HTML document to the browser associated with that VS client. The answering HTML document contains the IP address and port number of a VS server that will handle that particular scene. The Web browser then forwards the received server information to the VS client.

According to the Virtual Society architecture, the VS server is an independent program from the WWW server. Consequently, the Virtual Society does not support information exchange between the data on an HTML document and the objects inside a virtual world. Hence, the Virtual Society is an extension of the networked virtual reality system to the Web environment rather than merges the networked virtual reality system into the Web architecture.

German National Research Center for Information Technology (GMD) proposed another type of web-based multi-user system by defining new nodes for VRML standard [12]. Different from the Virtual Society project, the multi-user system of GMD does not require a multi-user server to maintain the consistency among distributed objects. Instead, after a VRML scene file has been downloaded from a Web server, the browser introduces itself as a user to the server. The Web server responds to this new user with an embodiment file of all existing players, along with a multicast address and a port number. The multicast address refers to the IP address of the Web server, and the port number denotes an existing virtual world. Hence, during the course of interaction, each participant sends its status change to the specified port on the Web server and listens to the update information of other users from another port representing a multicast channel.

Since each individual client sends the update messages to the specified port and the Web server redistributes the messages through the multicast channel, an extended HTTPD server is designed to receive update information to monitor the status of the virtual world. If the HTTPD server does not receive an update message from a client for a designated period of time, a time-out mechanism on the HPPD server will issue a quit message to the multicast channel for that client.

To achieve the consistency control of shared objects, GMD proposed a so-called Active Lock mechanism and two extension nodes, Interaction node and Behavior node, for the VRML standard. Although the research of GMD was concentrated on extending the VRML standard for the multi-user interaction, issues of seamless integration discussed in the next subsection still require further studies.

The Scalable Platform for Large Interactive Networked Environment (SPLINE) from Mitsubishi Electric Research Lab (MERL) used yet another approach to integrate the multi-user virtual environment with the Web [15]. SPLINE is essentially a network-based multi-user virtual reality system that uses Interactive Sharing Transfer Protocol (ISTP) as its communication protocol [16]. ISTP incorporated several simple HTTP protocols to enable the HTTP server to participate the multi-user interaction. With the help

of ISTP, the SPLINE system provides a Web-like virtual environment. However, it is not a web-based multi-user system.

Other commercially available web-based multi-user systems are OnLive! Community browser [17] from OnLive! Technologies Inc., OZ Virtual [18] from OZ Interactive Inc., CCpro[19] from Blaxxun interactive Corp. and V*Realm Multi-User browser [20] from Integrated Data Systems, Inc. All of the above browsers adapt the VRML specification as their virtual scene file format, and each system uses its proprietary technique to provide interactivity among distributed users. In addition, none of the above browsers are fully integrated with the traditional HTML browser. Active World browser [21] from Worlds Inc. is another kind of multi-user browser that is not a VRML browser, but instead, it merges the traditional HTML browser into its own 3D VR browser.

2.3 Distributive Interactive Simulation (DIS)

Although VRML brought virtual worlds into the Web environment, the only standardized communication protocol for multi-user interaction over the Internet is the Distributive Interactive Simulation (DIS) standard [6]. The DIS standard is originally designed for peer-to-peer simulation and aims at military training simulation; however, its concept has been adapted by other distributed virtual reality systems [11, 9]. In addition, the Protocol Data Unit (PDU) and the Dead Reckoning (DR) model have been proven to be two useful techniques to support the web-based virtual reality system [22]. Hence, an overview of the DIS standard is given here before further discussion.

The goal of the DIS standard is to link the interactive, free play activities of people in an operational exercise to form a time- and space-coherent synthetic world. This synthetic environment is created through the low latency (100 to 300 milliseconds) exchange of data units between distributed, computationally autonomous simulation applications. These computational simulation applications may be presented in one location or distributed geographically. Two important techniques have been defined in the DIS standard: the PDU and the DR model.

Since the DIS standard requires each participated host to have the replicated homogeneous world database, the PDU is the data unit that defines the information exchanged among simulation hosts through the network. The PDU contains the information of simulated entity status and the type of interaction that took place in an operational exercise. It also defines the data format for the simulation management to monitor the simulation process. The DIS defines 27 types of PDUs, which are organized into six protocol families: entity information/interaction, warfare, logistics, simulation management, distributed emission regeneration and radio communication. The Entity State PDU, that defines the information required to communicate an entity's state to other entities, is the most important PDU among them.

Furthermore, since the purpose of the DIS technique is to link the simulated entities distributed over geographically separated sites, a so called Dead Reckon model is used to estimate each entity's position and orientation based upon the pervious information. The objective of this Dead Reckon model is to reduce traffic load on the network when maintaining the spatial consistency among simulated entities [23]. For each simulated entity, the Dead Reckoning model uses its previous updated information to predict its next position and orientation, called Dead Reckon values. A host uses the Dead Reckon values to move distributive simulated entities before their actual postures are received from the network. In addition, for the simulated entity that is controlled by a host, its Dead Reckon value is calculated along with its actual position and orientation. With the Dead Reckoning technique, it is not necessary for a host to send an Entity State PDU of a simulated entity about every change in position and orientation that occurs over the time. Only when the Dead Reckon value of an entity is different from its actual posture by a predetermined threshold, a new Entity State PDU for that entity is actually sent onto the network. When other hosts in the same simulation receive this Entity State PDU, they will correct the posture of this entity to the updated value and resume its Dead Reckoning calculation from this new posture.

In other words, this approach specifies when the status change of an entity must be transmitted by a host to other participants. It also specifies how the state of an entity is estimated by each host before a status change information of that entity arrives. Thus, a participant can send less information to the network with the Dead Reckoning model.

3. The Petri Net Model for the Web-based Collaborative Virtual Environment

The survey from the previous section shows that, due to the constraint imposed by the Web architecture, the mechanism to support the

multi-user interaction on the Web is more complicated than what is on the network-based collaborative virtual environment. In addition, the previous survey also shows that the key to provide multi-user interaction on the Web architecture is the ability to seamlessly integrate with the existing Web architecture. The goal of this paper is to model the workflow of a seamlessly integrated web-based multi-user environment so that the researcher can easily design such a system on the Web. Hence, in the following subsections, the definition of the seamless integration is introduced first and followed by the proposed Petri Net model to capture the workflow on the Web architecture.

3.1 Characteristics of the Seamless Integration

A multi-user interaction virtual reality system on the Web architecture must provide mechanisms for the multi-user server to "remember" the information of the registered participants and to process the messages communicated among them [12, 13]. However, in order to take full advantage of the Web environment, the supported multi-user virtual reality system must be seamlessly integrated with the Web architecture. For a distributed multi-user system to be seamlessly integrated with the Web environment, it must have the following characteristics:

1. The user can download a scene file from any Web server with the HyperText Transfer Protocol (HTTP). That is, the user can select a scene file from any supported HTML document and access that virtual world by double clicking the mouse button.
2. The multi-user system must provide the hyperlink feature to retrieve various media resources that are supported by the Web environment. With the help of this feature, the user of the network-based virtual environment can easily access any media resource, such as a video, sound or image file, hyperlinked by objects inside the virtual world.
3. The multi-user system must be able to handle information exchange between the data on an HTML document and an object inside a virtual world. That is, the user can easily control an object inside the virtual world by filling data into forms on an HTML document. Similarly, the contents of an HTML document can be modified by the status change of an object inside the virtual world. Since an HTML document provides a more convenient way to display information, this feature is very important for a multi-user system to be completely integrated into the Web architecture. The most obvious application is to support a distributed 3D war-gaming environment [24].
4. The multi-user server itself is an add-on function of an existent Web server. This characteristic makes the multi-user interaction a part of the WWW services and allows the multi-user server to easily access the database provided by the Web server.
5. The user can directly enter a virtual world from a Web server from which the scene file is downloaded. With this feature, since the Web server takes the role of the user's login process, the multi-user server can be easily replaced and upgraded. In addition, the fault tolerance and the load balance features among the servers of the multi-user virtual reality system can also be easily implemented.

In summary, the seamless integration implies that the user can download a virtual scene file from any supporting Web server and navigate to other virtual worlds, which is managed by other Web servers without the awareness of the user. At the same time, the user can fully explore the services provided by the Web environment.

3.2 The Petri Net Model

Before we begin to introduce the Petri Net model for the multi-user collaboration on the Web, the basic concept of the Petri Net is given as follows. The Petri Net was originally proposed by C. A. Petri[25] which attempts to develop a formal methodology to describe and analyze a system behavior. The Petri Net model is a graphical and mathematical modeling tool which is especially useful to capture the synchronization characteristic among modules of a system. With the help of the netted representation by the Petri Net, the researcher can easily discover the potential problem of a running system and adjust its design to maintain the validity of this system.

A Petri Net model can be formally denoted as a 5-tuple, PN = (P, T, F, M_o) where:

- $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places.
- $T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions. Most importantly, P and T must satisfy the properties of $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$. That is, at least one of these two sets P and T must be nonempty.
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation) that network places and transitions. That is, $(P \times T)$ represents the set of arcs that flow from places to transitions whereas $(T \times P)$

is the set of arcs flowing in opposite directions.

- M_o: $P \rightarrow \{m_0, m_1, m_2, \ldots, m_m\}$ is the set of initial marking of each place. For the definition of the Petri Net model, m_{ij} represents the token number on place p_j at time *i* and a token can be a resource of a system or control of a program.

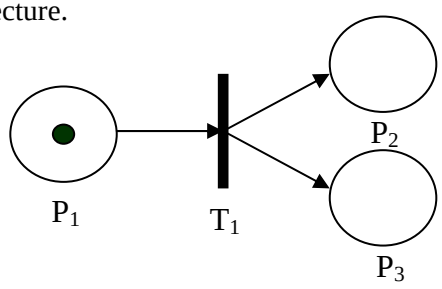
According to the definition of the Petri Net, Figure 1 depicts the case of a Petri Net model before and after a fire event. Circles P_1, P_2 and P_3 in Figure 1 represent places of this model and bar T_1 represents a transition.

The generic components of a Petri Net include a finite set of places and a finite set of transitions. A Petri Net is a finite bipartite graph that places are netted with transitions, which in turn are connected to output places. The distribution of tokens over places is called a marking of the net. A transition may enable or fire when each of its input places contains at least one token. The firing of a transition results in removing tokens form their input places and adding to output places via transition. A marking represents the state of a system, which is removed from its place when a transition fired and a new marking is then generated to the output places of this transition.

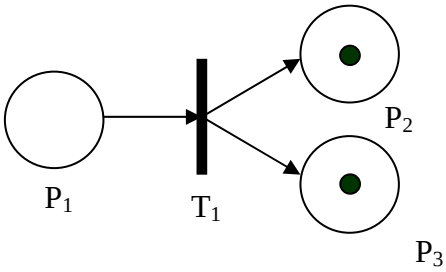
3.3 The Petri Net model of a web-based multi-user collaboration system

A web-based multi-user system can be classified into client and server sites. The client site is generally composed of four modules: "Input unit", "Output unit", "Interaction broker" and "Network interface" as illustrated in Figure 2. For the server site, in order to focus on the workflow of the multi-user interaction on the Web, we model the server site by only two modules, Client manager and Network interface, without loosing the generality of this model.

With the given infrastructure for the Web-based multi-user system, we can then begin to model its workflow by the Petri Net model. First, we assign the meanings of places and transitions to model the functional behavior of the system. According to the infrastructure depicted in Figure 2, we model the functional module of the system by the "transition" and the input parameter of each functional module as the "place". Secondly, we specify the necessary stages of the web-based multi-user virtual reality system that needs to be modeled. From the previous surveys, we know that a Web-based multiple participants virtual reality system should provide the following five stages to support multi-user collaboration on the Web architecture.



(a) A initial marking before transition firing



(b) A new marking after transition firing

Figure 1. The Petri Net in fire

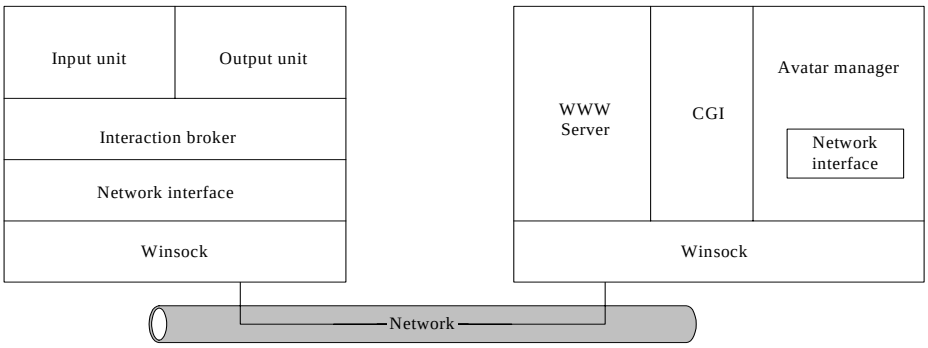


Figure 2. The infrastructure of a web-based multi-user system

- **The Initialization stage:** This stage is related to the first characteristic of seamless integration. For the web-based application, the user should be able to click and download a virtual world from a supported HTML document.
- **The Logging stage:** For an application to be seamlessly integrated into the Web architecture, the client should be able to log into the multi-user server from the Web server that stores the scene file. This stage is consisting with the fifth character as discussed in subsection 3.1.
- **User status exchange stage:** When the users are collaborating inside the virtual world, the spatial consistency within the virtual world is the essential issue to achieve this objective. This stage exchanges user status information to reach this spatial consistency criterion.
- **The chatting stage:** Chatting is another important feature for the collaborative virtual environment.

The Petri Net model of these five stages are then elaborated as follows:

3.3.1 The Initialization Stage:

In this stage, the client site accesses the HTML or 3D VRML file from the server site. As shown in Figure 3, this stage is started from the user activate the Interaction broker via Input unit which then sends message to the Web server from the Network Interface.

First of all, the user uses the mouse, Transitions t1.1, or keyboard, t1.2 system, to load a web page or a 3D VRML file. The main function of Transition t1.1 is to open and browse the HTML file in the local machine. Similar to the existing web browser, the supported browsing action should include backward, forward, stop and reload a HTML files. When the keyboard module is used, the "Identify File Name module"(Transition t2.1) is called to distinguish whether inputted URL is for a web page or a 3D VRML scene file? According to the result, either a "Download HTML file module" or a "Download 3D VRML file module" is activated to request respective file from the WWW server through the "HTTP protocol module". In Figure 3, the "Download HTML File module" is assigned as Transition t4.4 and the "Download 3D VRML File module" is Transition t4.5.

When a web page is replied from the WWW server, this received file will be forwarded to the "HTML parser" as specified as Transition t2.2, which will then be forwarded to the HTML

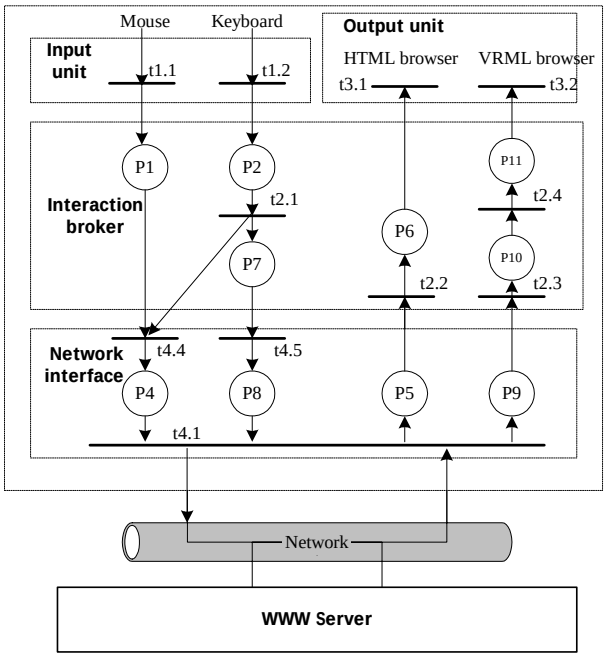


Figure 3. The Petri Net model for the Initialization stage

browser, i.e. Transition 3.1. On the other hand, if a 3D VRML is received from the Web server, it is first sent to the "VRML parser"(Transition 2.3) to parse and generate a virtual world for the "Scene Manager module"(Transition 2.4) and rendered by the "VRML browser"(Transition 3.2).

3.3.2 Login to the Multi-user World

After a virtual scene is downloaded and rendered on the client site, the user needs to login to the multi-user server in order to participate the virtual world. As shown in Figure 4, the user inputs his user name through the "User Name Input module" module(Transition t1.5). The "Registrant module" (Transition t2.5) is then called to calculate a Unique Identification Number (UID) based upon the inputted user name and other related information. This unique identification number is forwarded to the "Message Packer"(Transition t4.7) to pack into a registration packet and sent out to the Web server by the "Packet Sender module"(Transition 4.2).

Upon receiving the registration packet, the Web server forwards this information to the multi-user server by a CGI program. The "Registrar module" (Transition s1.2) on the multi-user server then authenticates the validity of the user. The result of authentication is then sent back to the client site through the "massage packer module"(Transition s2.4) and the "message sender module"(Transition s2.1).

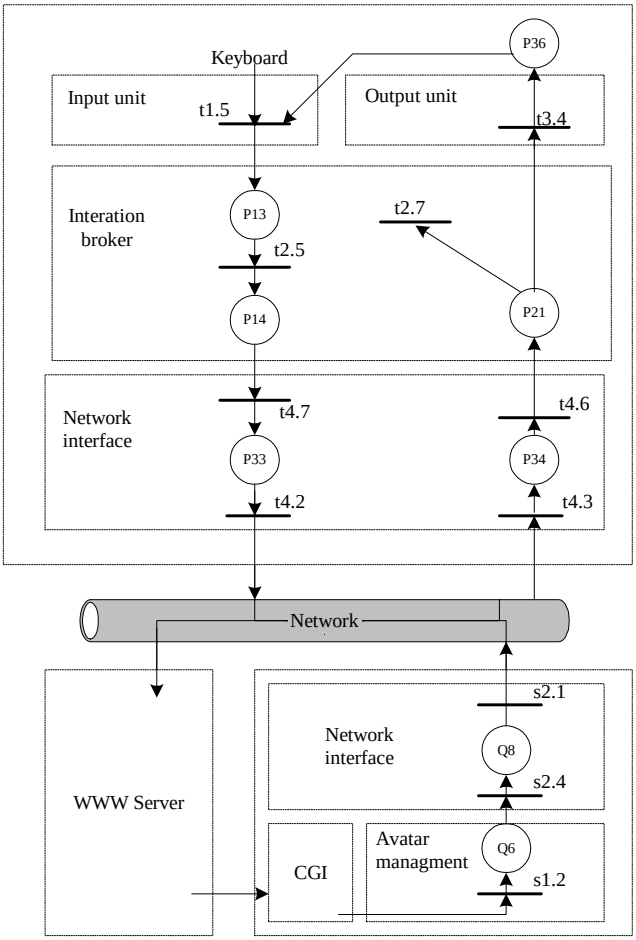


Figure 4. The Petri Net model for the Login stage

When the authentication packet is received by the "Packet Receive module"(Transition t4.3) on the client site, this packet is decoded by the "Packet Discharger"(Transition t4.6). If the user is a valid player, then the "Enable Trigger module"(Transition t2.7) is set to start up all the functions of the Interaction Broker. Otherwise, an error message will be displayed on the Output Window.(Transition t3.4)

3.3.3 User Status Flow:

The user status includes the avatar type selected and the position as well as the orientation of the avatar controlled by the user. As depicted in Figure 5, the "Movement Bar module"(Transition t1.3) provides an interface for the user to control the navigating speed and direction of the avatar and these control messages are forwarded to the "Scene Manager module"(Transition t2.4). In order to lessen the traffic load, we can reduce the number of packets sent out onto the network by employing a Dead Reckon technique [23]. The Dead Reckon technique is the method of employing the previous status of an avatar to predict its next status. If the discrepancy between these consecutive states is within a threshold value, then no update message is sent and the local site uses the predicted status as its current status to display. Hence, the "Dead Reckon module"(Transition t2.6) is looping with the "Scene Manager module"(Transition t2.4) for the image rendering by the "3D Render module"(Transition t3.2) and to decide if a new message is required to be sent out by the "Message Packer"(Transition t4.7) and by the "Packet Sender module"(Transition 4.2). In addition, the user can use the "Avatar Select Module "(Transition t1.4) to change his roles inside the virtual world.

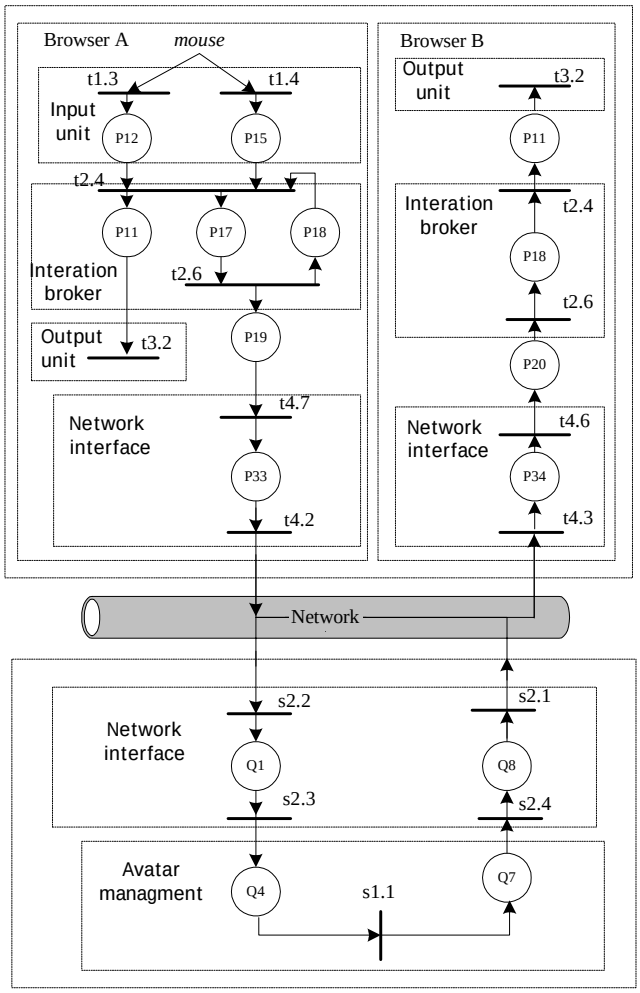


Figure 5. The Petri Net mode for the user status exchange stage

When the Multi-user server receives the update message from the client site by the "Packet receiver"(Transition s2.2), this message will forward to the "Motion Tracker module"(Transition s1.1) for the spatial consistence maintenance by the "Packet Discharger"(Transition s2.3). The tracking result is

then casted to all other players within the same virtual world from the "Packet packer module"(Transition s2.4) and the "Packet sender module"(Transition s2.1). When a client receives this update message from the Multi-user server, this message is then forwarded to the "Dead Reckon module"(Transition t2.6) as well as to the "Scene Manager module"(Transition t2.4) to update the respective avatar status.

3.3.4 Chat Room

Chatting among clients is an important function for a collaborative virtual environment. There exist two methods to provide chatting over the Internet. One is to transmit voice message and the other is simply by the interactive exchange of the textual message. Since the voice transmission technique is much more complicated and requires enormous network bandwidth, most of the existing collaborative virtual reality systems only provide the textual message exchange. Hence, we also model the textual message transmission here.

As shown in Figure 6, the user uses the "Chat Input box" (Transition t1.6) to input his textual message which will be interactively shown on the "Chat Out Windows"(Transition t3.4). At the same time, the "Listener module"(Transition t2.9) will multicast this textual message to the other users through the "Message Packer"(Transition t4.7). The "Listener module" maintains a "Chat room", which is essentially a client table that joins this chat room, to instruct the "Message Packer"(Transition t4.7) to forward this message. Notice that, according the chatting protocol [26], this textual message is sent out by the "Packet Sender module"(Transition 4.2) using the UDP protocol.

4. Implementation of the SharedWeb System

4.1 The architecture of the SharedWeb system

In order to verify the model present in the above section, a web-based collaborative virtual reality was implemented, called the SharedWeb system, in Tamkang University. Based upon the previous definition of seamless integration, the SharedWeb system was implemented to provide a natural extension of the Web environment into a multi-user virtual environment. The infrastructure of the SharedWeb system is shown in Figure 7.

The nodes of the SharedWeb system can be classified into two types: the server site and the browser site. The browser site is composed of four

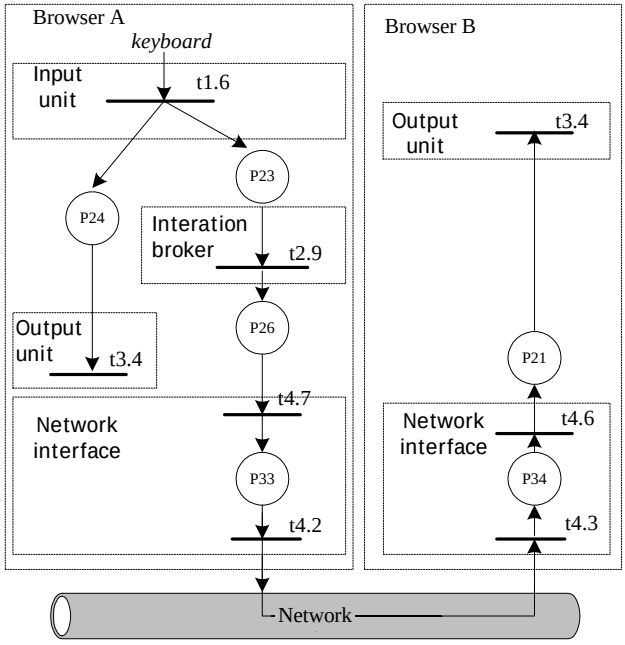


Figure 6. The Petri Net model for the textual chatting stage

modules: Multiple Participants Interface, 3D Render Engine, Chat Phase, and WWW Homepage Viewer modules. The server site is designed as an add-on application of the existing Web server with two modules: a CGI program and the SharedWeb server. Each of these modules is responsible for the tasks described below:

- **The Browser Site**

The Multiple Participant Interface module runs on top of Winsock to manipulate the interaction between the browser and other players. This module is the kernel of a SharedWeb browser. Its main functions are to calculate browser's UID, encapsulate and de-capsulate message packets, transmit/receive data to/from the network, and coordinate messages among 3D Render Engine, Chat Phase, and WWW Homepage Viewer modules.

The 3D Render Engine module provides a visual window for the user to explore the virtual world. When a browser retrieves a scene file from the Web server, it is an ASCII file. The scene file is then translated into a virtual world database recognized by the 3D Render Engine module.

The Chat Phase module is designed to support text communication among distributed players. This module enhances the interactions among distributed participants and, hence, broadens the applicability of the SharedWeb system. Without this module, the shared virtual world will be a wordless world that easily makes the user bored

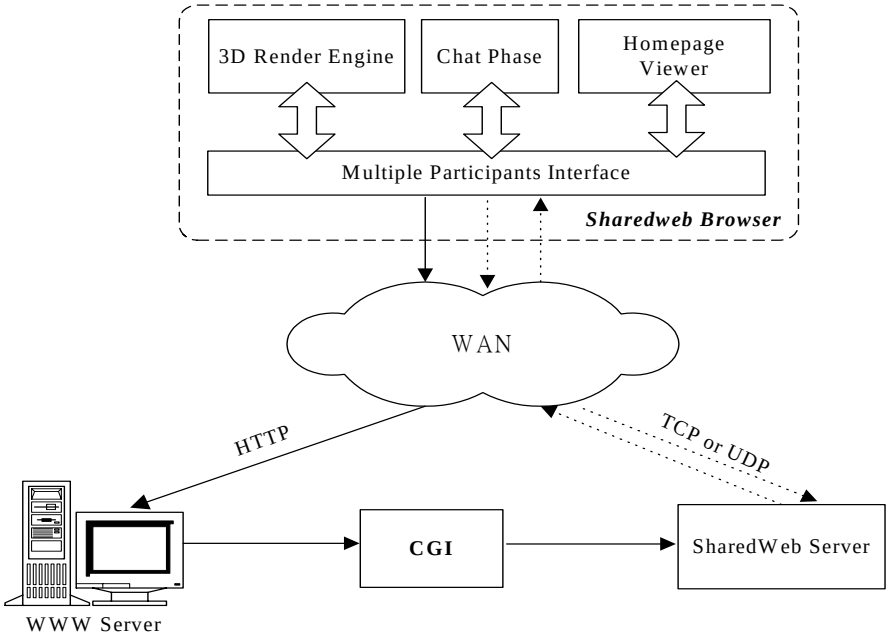


Figure 7. The architecture of the SharedWeb system

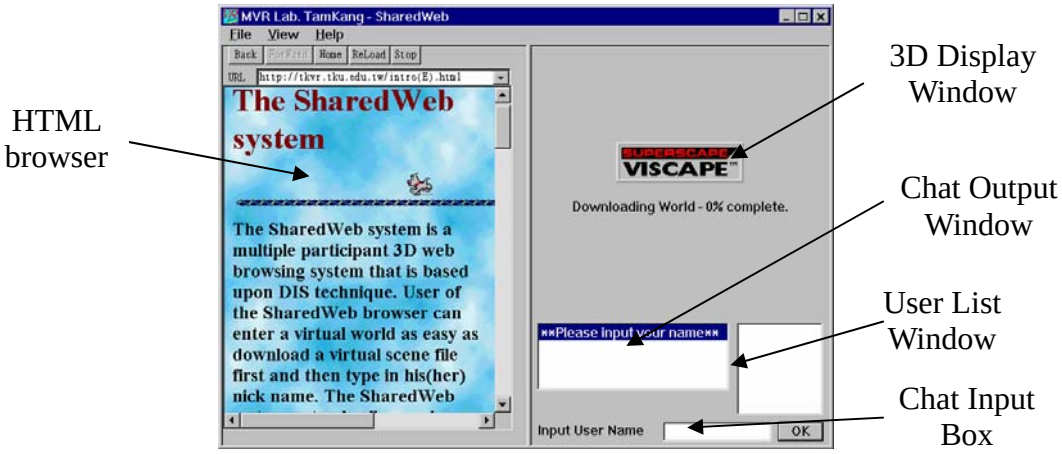


Figure 8. Interface of the SharedWeb browser

when playing with the SharedWeb browser.

The WWW Homepage Viewer module is a standard HTML browser. By seamlessly integrating this module into the SharedWeb browser, a user can enter another virtual world by double clicking a hyperlink provided on a homepage or inside the virtual world

- **The Server Site**

The Web server is a gateway for a SharedWeb user to enter a virtual world. A user only needs to download a scene file from this Web server and logs in a nickname to enter a virtual world. After the Web server receives the login information from a browser, it passes the received user information to the SharedWeb server through the CGI program.

The SharedWeb server module is the kernel of the entire SharedWeb system. This program is responsible for mediating the information exchange among the browsers in the same virtual world. All the client information is sent to the server, and the server routes the received information to the browsers in the same virtual world, with the exception of the Chat function among the users.

4.2 The Implementation

The SharedWeb system was implemented on Microsoft Windows platform using Visual C++. The SharedWeb server is run on Windows NT Server with an embedded WWW server. In addition, for the sake of flexibility and

extendibility, the SharedWeb browser was implemented as a standalone browser at this moment. The SharedWeb browser will be implemented as a plug-in application for the Netscape Communicator or the Internet Explorer in the near future. The interface of the SharedWeb browser is as shown in Figure 8.

As displayed on the upper-right window of Figure 8, Viscape software from Superscape Inc. [27] is used as the 3D Render Engine. The lower-right window is the Chat Output Window that will display the text messages that are communicated among distributed users, and the user can type in his message from the Chat Input Box. When the browser is initiated, the Chat Input Box is originally labeled as "Input User Name" and is used for the user to type in his nickname to login the downloaded virtual world. Before the user logins a virtual world, the chat function and the multi-user interaction feature are disabled. After the user has successfully logged in a virtual world, the Chat Input Box will be activated automatically and all of the users' names residing in the virtual world will be listed in the User List Window. Furthermore, a "Send URL" button will emerge at the same time, as shown in Figure 9.

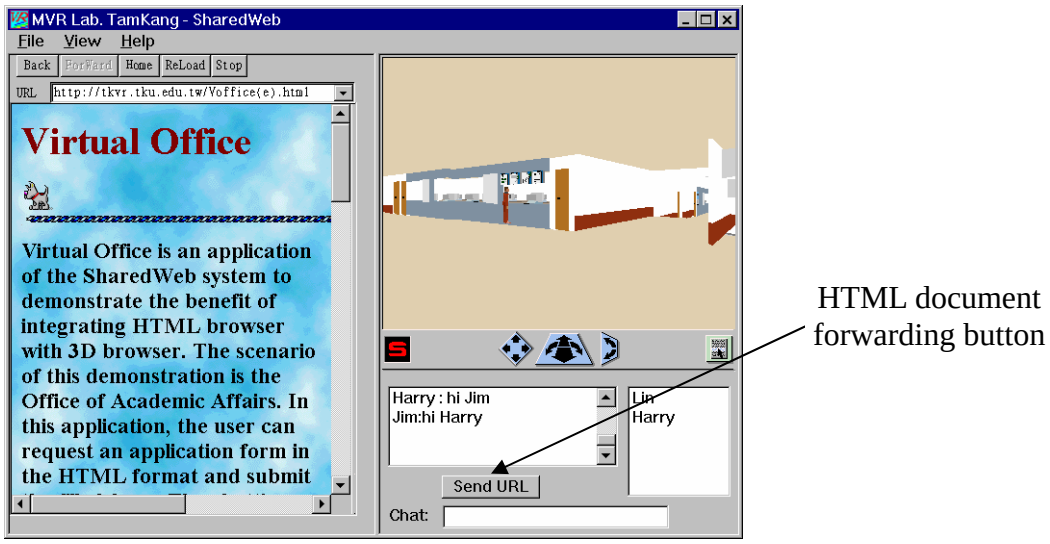


Figure 9. URL button for forwarding HTML document

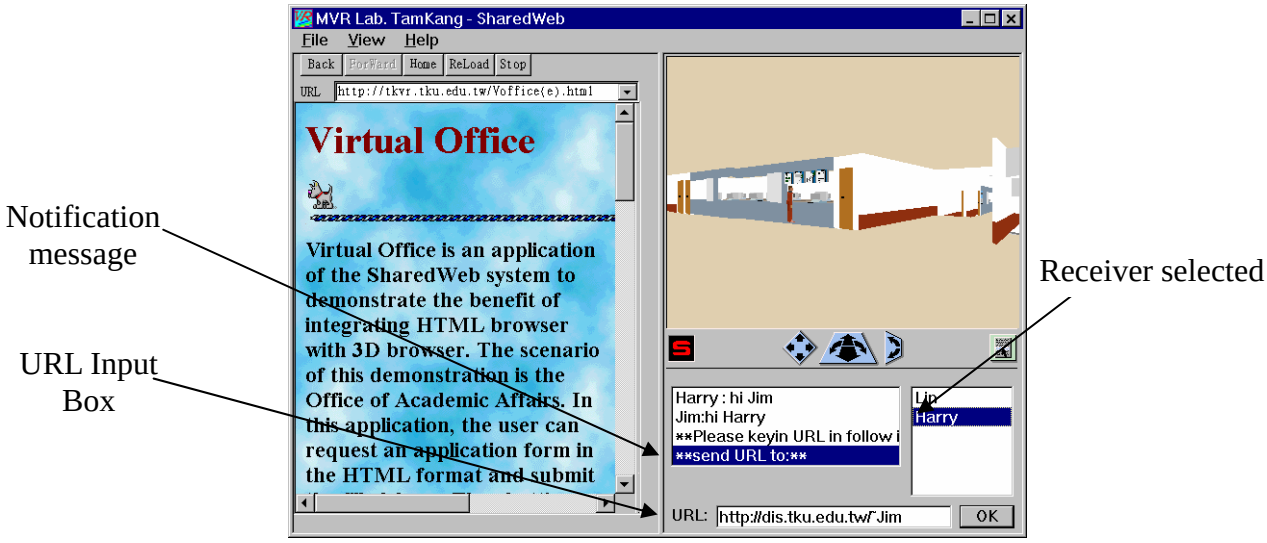


Figure 10. Snapshot after the "Send URL" button is pressed

The "Send URL" button is a special function for the user to send a hyperlinked resource, such as an HTML document or audio clip, to other users. This function is another feature for the SharedWeb system to be seamlessly integrated into the WWW environment.

After the "Send URL" button is pressed, it will disappear again and a highlighted message will be displayed in Chat Output Window to notify the user. In the meantime, as shown in Figure 10, the label of the Chat Input Box is changed to "URL:" for the user to input the URL address of a web resource, such as an HTML document or audio clip. The sender selects the receiver(s) by clicking one or more names from the User List Window. By clicking the OK button, the specified network resource will then be automatically forwarded to the receiver(s).

5. Demonstrations

To fully explore the features provided by the SharedWeb system, different virtual worlds were constructed. The following scenarios are part of the virtual worlds currently supported in the SharedWeb environment.

5.1 Virtual Campus

The virtual campus is a scaled-down duplication of the main campus of Tamkang University, Taiwan. The purpose of this virtual campus application is to investigate the possibility of designing a virtual University over the Web environment. This application provides a new way for the students to visit the campus without leaving their desk. Inside this virtual campus, the students

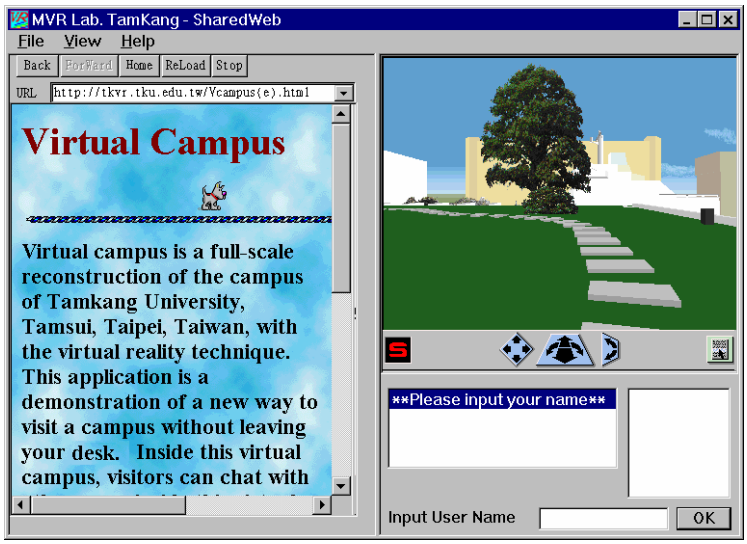


Figure 11. The snapshot of the virtual campus after it is downloaded.

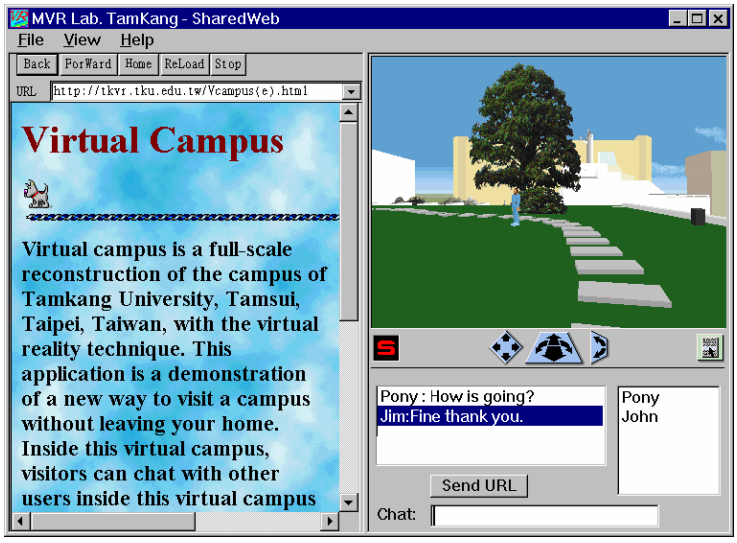


Figure 12. Snapshot of the virtual campus after the user, "**Jim**", has logged in.

can chat with each other while navigating the virtual campus. In addition, each building inside this virtual campus has a hyperlink to another virtual world that models the interior of that building. If a building has many storeys, it will contain hyperlinks to more than one virtual world and each floor is modeled as a virtual world. Figure 11 shows the snapshot when the user downloads the scene file of the virtual campus. In addition, a brief introduction of this virtual world will also be automatically downloaded as an HTML document at the same time. However, before the user logins a virtual world, he is alone.

Figure 12 is an illustration of the virtual campus from the view of user **"Jim"**. When the user logins to the virtual campus, the names of all users that are already inside this virtual world will be displayed on the User List window. This particular scene includes three users with their names, except the observer "**Jim"**, are displayed on the User List window. Each user can navigate the virtual campus by clicking and dragging the control panel provided by the Viscape software, and the other user inside the virtual campus will observe the movement of his represented avatar. Figure 12 also shows the result of the user **"Jim"** greeting to another player **"Pony"** through the Chat Phase module.

When a user wants to chat with other participants, he can input his message into the Chat Input window and this message will be displayed on the Chat Output Window of each participant's browser. Notice that, before the user enters a virtual world, the Chat Input Box is labeled as "Input User Name" as shown in Figure 8. After the user has successfully logged in the virtual campus, the label of the Chat Input Box will be automatically changed to "Chat:".

5.2 Virtual Office

The virtual office is a hyperlinked virtual world when the user clicks on the Administrative Building of the virtual campus. The virtual office application is an effort to study the possibility of performing some of the university's administrative tasks online.

Since the SharedWeb system is an integration of the virtual reality system with the HTML browser, a user can use the HTML document to fill the application form, says, for a new student ID. As shown in Figure 13, various application forms are hyperlinked by pictures on the wall inside this virtual office. The user can click one of the pictures to download an application form to the HTML browser. Since the application form is implemented as an HTML document, different types of the application form can be easily designed and modified. Hence, a virtual shopping mall can be easily implemented with the help of this integrated HTML browser as well.

Figure 14 shows that, when the student has filled the application form and clicked the "Send" button on the HTML document, the filled information will be sent to the WWW server for processing. After the server has successfully received this information, the server will reply an acknowledgement message to the client.

6. Conclusion and Future Works

This paper presents a Petri Net Model of the We-based collaborative virtual environment. This model presented in this paper comprehensively reveals the interaction between the client and the server during the initialization stage as well as the login stage. In addition, the interactions among clients and multi-user server during the run-time stage are also discussed in this model. Establishing a web-based collaborative virtual environment over the Internet is a popular research in the recent years. Designing a collaborative virtual reality system on the existing Web architecture is a growing interest and a technical challenge. Compared to the legacy network-based multi-user virtual reality system, the Web-based collaborative virtual reality system requires the collaborative system to be fully integrated with the existing Web architecture. This particular characteristic is referred to the feature of seamless integration and it is completely discussed in this paper.

In order to verify the presented model, the architecture of a web-based collaborative virtual reality system is designed and implemented. This implemented web-based collaborative system is called SharedWeb system which means a shared web environment can be easily constructed with the implemented system. Under the SharedWeb system, the existing Web server remains intact while supporting the multiple participant interactions. Several demonstrative virtual worlds were also designed to fully explore this useful feature. Other applications such as virtual shopping mall and virtual classroom are currently under studies.

Although the current SharedWeb implementation successfully demonstrates the concept of seamless integration with the Web environment, more research is required to improve the multi-user interaction. For example, the consistency control of the shared objects is an

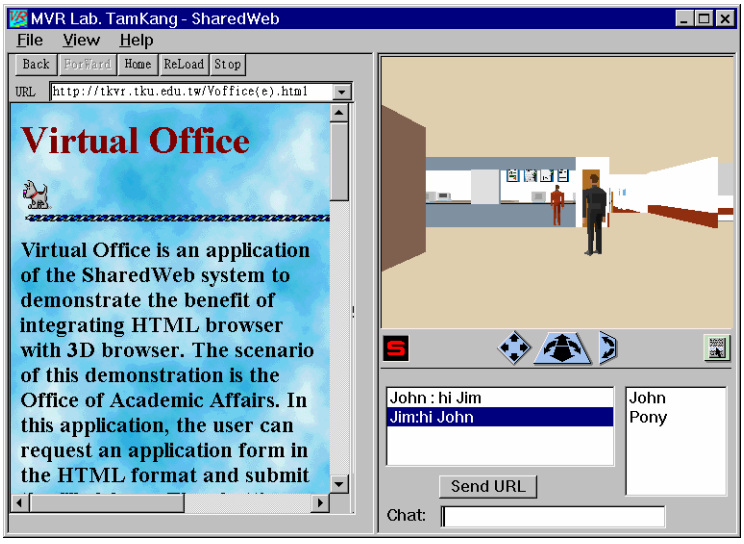


Figure 13. Pictures on the wall of the virtual office representing various application forms

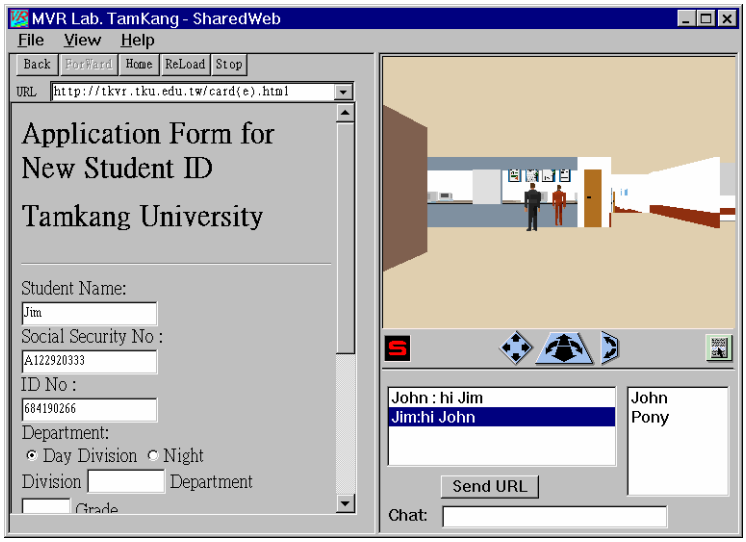


Figure 14. The snapshot after the user has downloaded and filled an application form

important function for the computer-supported cooperative work (CSCW) type of applications. Spatial culling technique to reduce the network bandwidth requirement is another important feature that is currently under investigation. Finally, a server cluster architecture, that would organize the virtual world database into a client-server architecture and thus scales up the number of participants, is an undergoing investing topic. This function would also provide fault tolerance on the server site.

References

[1] Berners-Lee, T., et al., "The World-Wide Web", Communication of the ACM, Vol. 37, No. 8, pp. 76-82, August (1994).

[2] "The Virtual Reality Modeling Language Specification Ver. 2.0", available on line at <http://vrml.sgi.com/moving-worlds/spec>, August (1996).

[3] Macedonia, M. R. and Zyda, M. J., "A Taxonomy for Networked Virtual Environments", IEEE Multimedia, Vol. 4, No. 1, pp. 48-56, January-March (1997).

[4] Calvin, J., et al., "The SIMNET Virtual World Architecture", Proc. of the IEEE Virtual Reality Annual International Symposium, pp. 450-455, September (1993).

[5] Macedonia, M. R., et al., "NPSNET: A Multi-Player 3D Virtual Environment Over The Internet", ACM SIGGRAPH Special Issue on 1992 Symposium on Interactive 3D Graphics, (Cambridge, MA), pp. 93-94, (1995).

[6] “Standard for Distributed Interactive

Simulation (DIS) – Application Protocols", ANSI/IEEE Std, pp. 1278-1993, March (1993).
[7] Funkhouser, T. A., "RING: A Client-Server System for Multi-User Virtual Environments", ACM SIGGRAPH Special Issue on 1992 Symposium on Interactive 3D Graphics, Cambridge, MA, pp. 85-92, (1995).
[8] Capin, T., et al., "Virtual Human Representation and Communication in VLNET", IEEE CG&A, Vol. 17, No. 2, pp. 42-53, (1997).
[9] Hagsand, O., "Interactive Multi-user VEs in the DIVE System", IEEE Multimedia, Vol. 3, No. 1, pp. 30-39, Spring (1996).
[10] Singh, G., et al., "BRICKNET : Sharing Object Behaviours on the Net", Proc. of VRAIS'95, IEEE Computer Society Press, pp. 19-25, March (1995).
[11] Greenhalgh, C. and Benford, S., "Massive, A Collaborative Virtual Environment for Teleconference", ACM Trans. on Computer-Human Interaction, Vol. 2, No. 3, pp. 239-261, (1995).
[12] Broll, W., "VRML : From the Web to Interactive Multi-User Virtual Reality", Proc. of the GI Workshop on Modeling - Virtual Worlds - Distributed Graphics, Bad Honnef/Bonn, Germany, Also available at <http://orgwis.gmd.de/projects/VR/vrml/papers/MVD95.ps>, November (1995).
[13] Honda, Y., Matsuda, K., Rekimoto, J. and Lea, R., "Virtual society: extending the WWW to support a multi-user interactive shared 3D environment", Proc. of VRML'95, San Diego, CA., Also available at <http://www.csl.sony.co.jp/person/rodger.html>, August (1995).
[14] "Community Place Browser", available on line at <http://www.sonypic.com/vs>
[15] Waters, R. C., et al., "Diamond Part and Spline: Social Virtual Reality with 3D Animation, Spoken Interaction, and Runtime Extendability", Presence, Vol. 6, No. 4, pp. 461-481, August (1997).
[16] Waters, R. C., Anderson, D. B., Schwenke, D. L., "The Interactive Sharing Transfer Protocol Version 1.0", available on line at <http://www.merl.com/reports/index.html/TR97-10>
[17] "OnLive! Community Browser", available on line at <http://www.onlive.com>
[18] "OZ Virtual", available on line at <http://www.oz-inc.com/ov>
[19] "Blaxxun Community Client", available on line at < http://ww3.blacksun.com/ >
[20] "V*Realm Multi-User Browser", available on line at <http://www.ids-net.com/ids/vrealm>
[21] "Active Worlds", available on line at <http://www.activeworlds.com/>
[22] Huang, J. Y., Wang, F. B., Hsu, W. H. and Chen, J. F., "Usage of DIS Technique to Create an Interactive WWW Environment", 14th DIS Workshop, Orlando, Florida, pp. 201-210, March (1996).
[23] Lin, K. C. and Schab, D. E., "The Performance Assessment of the Dead Reckoning Algorithms in DIS", SIMULATION, pp. 318-325, November (1994).
[24] Huang, J. Y., Chang, J. L., Li, C. W. and Lin, K. C., " Design of a Multiple Participant 3D War Game Environment over WWW", SPIE's 12th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Control, Orlando, FL, April (1998).
[25] Peterson, J. L., "Petri Net Theory and the Modeling of Systems," Englewood Cliffs, NJ: Prentice-Hall, Inc., (1981).
[26] Oikarinen, J. and Reed, D., "Internet Relay Chat Protocol", Internet RFC #1459, May (1993).
[27] "VRT for Windows - User Guide", Superscape Inc., UK, (1996).

Manuscript Received: Nov. 16, 2000
Accepted: Dec. 27, 2000