Hypermedia and Networking in the Development of Large-Scale Virtual Environments

Michael J. Zyda*, Chuck Lombardo, and David R. Pratt
Naval Postgraduate School
Department of Computer Science
Monterey, California 93943-5100
zyda@trouble.cs.nps.navy.mil
*contact author

Abstract

Current virtual environment systems are primarily geometric, i.e. they consist of worlds whose objects are constructed of polygons that are occasionally textured. The majority of the current environments are also little more than sophisticated single system or low connectivity (one or two workstation) "fly-through" systems, direct descendants of the traditional flight simulator. As virtual environments evolve, there will be a significant demand for more advanced information integration and for more advanced connectivity. One information integration capability that will have to be added into virtual environments is hypermedia data visualization, i.e. audio, and compressed video, with embedded links. Hypermedia consists of non-sequential media grouped into nodes that are linked to other nodes. If we embed such nodes into say a building of our virtual world, the node can be accessed and audio or compressed video containing vital information on the layout, design or purpose of that building can be displayed, along with historical information. Besides information node selection via mouse, we also expect audio and video "landmines", i.e. audio and video displays triggered and displayed as the virtual world is navigated.

Advanced connectivity is also one of the more important trends in the development of virtual environments. As we move from Ethernet to ATM, and move from 250 player to 10,000 player systems, the technical issues of distributed world models, model consistency and real-time scene management become significantly harder.

In this paper, we outline the issues involved in large scale networking and hypermedia integration and how those research questions are being addressed by the NPSNET Research Group of the Naval Postgraduate School.

Keywords: virtual worlds, hypermedia, networked virtual worlds, and virtual world software.

Virtual Environments Research Goals

The goal of the virtual environment (VE) research community is the construction of computer-generated, 3D worlds through which a human player can move and interact. The hardware technology used for such virtual environments is the computer graphics workstation, local and wide area networks, high resolution head-mounted displays, and novel 3D input devices.

The software technology for virtual environments is evolving rapidly. The goal of research in VE software is to generate a software bed for the rapid construction of fully-detailed, fully interactive seamless virtual environments. "Seamless" is the key concept, meaning that we can drive a vehicle across our terrain, stop in front of a building, get out of the vehicle, enter the building on foot, go up the stairs, enter a room and interact with items on a desktop. All without delay or hesitation in the system.

To build seamless systems, there is a lot of software progress that must be made. In this paper, we look at the issues involved in the networking and hypermedia software that must be developed. The reader is directed to [5, 6] for a more complete discussion of virtual environment software requirements.

Communications Software

Virtual environments communications software is software for passing changes in the world model to other players on the network and software for allowing the entry of previously undescribed players into the system. It is quite easy to grow out of a single workstation when constructing a virtual world, especially if one expects multiple players in that world. When we move to a networked environment, we are beyond issues of graphics and interface software and into a much more complicated system.

When we move into a networked environment, we need to consider issues of database consistency more closely than we do in the single workstation world. We end up needing a standard message protocol between workstations that communicates changes to the world. For *small* systems, we need to make sure that all players on the network have the same world models and descriptions as time moves forward in the virtual environment "action".

What do we mean by a small number of players? We define low numbers of networked players to be between 250 - 500 players. The current SIMNET system looks like this and uses Ethernet and T1 links. Each node in the virtual world has a complete model of the world.

We define high numbers of networked players to be systems with 10,000 to 300,000 players. The Defense Modeling and Simulation Office (DMSO) wants this. For systems of this complexity, we can no longer afford to propagate complete models of the world but must start thinking about rolling in the world model just as aircraft simulators roll in terrain. Are many people thinking about this now? No, but perhaps we can see at least an abstraction that might be relevant in the work of Gelernter entitled "Mirror Worlds" [1]. In "Mirror Worlds", we have the notion of information tuples (think distributed blackboard), and tuple operations: publish, read and consume. Such an abstraction allows flexibility in communicating any type of information throughout a large, distributed system. This flexibility is just what we need in constructing large virtual worlds. While the abstraction looks appropriate, efficient and real-time implementations are an open research problem.

Hypermedia Integration

Hypermedia integration software is the additic of hypermedia data (audio, compressed vide with embedded links) into our 3D geometrical described virtual world. The idea behind hyperm dia integration is to combine virtual world techno ogy with hypermedia technology by embeddir hypermedia nodes in the virtual world. Hyperm dia consists of non-sequential media grouped in nodes that are linked to other nodes. If we embe such nodes into say a building of our virtual worl the node can be accessed and audio or compresse video containing vital information on the layou design or purpose of that building can be di played, along with historical information. Suc nodes will also allow us to make a search of a other nodes and find related objects elsewhere i the virtual world.

We also envision hyper-navigation. Hyper-nav igation involves the use of nodes as markers th can be traveled between, either over the virtual te rain at accelerated speeds or over the hypermedi "links" that connect the nodes. Think of rabb holes or portals to information or informatio places.

We also envision hypermedia authoring. In au thoring mode, the computer drops nodes as a gam is played. After the game, the player can trave along these nodes (which exist not only in spac but also in time, appearing/disappearing as tim passes) and watch a given player's performance i the game.

Now that we have described the issues in gen eral with respect to virtual environment network ing and hypermedia software, let's now look at th NPSNET virtual world where some of these issue are being addressed.

The NPSNET Networked Virtual Environment

The NPSNET system is a low-cost, 3D visua simulator for virtual world exploration and exper imentation.The goal of the project is to develop basic virtual world shell that allows one to visi any area of the world for which a terrain databas is available and to interact with any interactive o autonomous players found "in the system". NPSNET is networked, i.e. we can put one interactiv player at each graphics-capable workstation on th network (for however many workstations w have) and as many autonomous players as our computational and network resources allow.

The goal of the NPSNET project is to construct a low-cost visual simulator/virtual world explorer interoperable with the DARPA SIMNET system and the follow-on DIS networking standard [2 - 4]. One of the tenants of the NPSNET project is to construct it on commercially available graphics workstations, in particular the Silicon Graphics, Inc. IRIS workstation in all its incarnations. We wanted to have full control of the source code and not only build a SIMNET-compatible system but also several significant extensions. We wanted to play with some autonomous player ideas we had and wanted to greatly improve the availability of the virtual world technology required to construct such a system. We wanted to build a virtual world system that we could put onto a tape and share with others without having to worry about whether it or the techniques involved were proprietary.

NPSNET Current Status

NPSNET is a family of virtual environment research systems, with variants numbered/named NPSNET 1 through 4, and Hyper-NPSNET. The core software technology advances with each system, with the newer versions supporting more advanced features as work on the research project progresses.

With NPSNET, we have explored the areas of real-time scene management, collision detection, aural cues, dynamic terrain (bridges that can be driven over/under, and craters/berms that can be driven over/into), physically-based modeling (aircraft flight modeling, ballistic projectiles), autonomous forces (CLIPS programs with DIS interfaces), constructive model integration (JANUS and EAGLE), and urban terrain (building walkthrough and vehicle motion).

NPSNET-4, the current developmental system, is being put together for a demonstration in the Tomorrow's Realities Gallery of SIGGRAPH '93. NPSNET-4 uses the Silicon Graphics, Inc. visual simulation toolkit Performer and is networked using the DIS 2.0.3 standard. The demonstration at SIGGRAPH '93 will contain a T1 link to the DSI-net (see below) and will have players at locations throughout the United States.

NPSNET-3 & Warbreaker

NPSNET-3 currently runs and was demonstrated in a networked test at the Institute for Defense Analysis in Alexandria, Virginia (Warbreaker/Zealous Pursuit Test, October through December 92). NPSNET provided a 3D display for the test and interoperated with an F-15 simulator at McAir, St. Louis, another simulator at Williams AFB in Arizona, and several other simulators on the local IDA network (Patriot launcher, JSTARS C3 Station, etc.). The long distance networking was provided by the Defense Simulation Internet (DSI-net), which consists of T1 links at approximately 150 sites throughout the US.

The software architecture of NPSNET-3 is shown in Figure 1. Basically, the core simulator communicates to the network via a protocol converter interface that sends/receives network packets asynchronously using both a "send thread" and a "receive thread". This allows the graphics display rate to be maintained while data is read/written in separate, lightweight processes.

NPSNET Networking Performance

We have done two major networking tests where we have measured the net. In our laboratory at the Naval Postgraduate School, we used a SIMNET Data Logger tape and SIMNET protocols to play an engagement at Fort Hunter-Liggett onto our local area network. In that test, we averaged approximately 100 messages per second for a period of 20 minutes coming from 270 different entities (interactive and autonomous players). The peak number of messages (Protocol Data Units - PDUs) was 210 PDUs in one second. The average packet length including the network header was 149 bytes. The simulation PDU traffic accounted for 50% of the network load but only 1% of the network bandwidth. The maximum one second burst was 2.3% of the bandwidth.

Our second measured network test was during the Warbreaker/Zealous Pursuit exercise. The goal of the exercise was to put players onto the DSI-net and see how the network performed and how the heterogeneous players involved in that exercise interoperated.

The average SIMNET PDU rate for that test was 142 packets per second. The network encryption devices had a limit of approximately 170 PDUs per second before the devices slowed down,

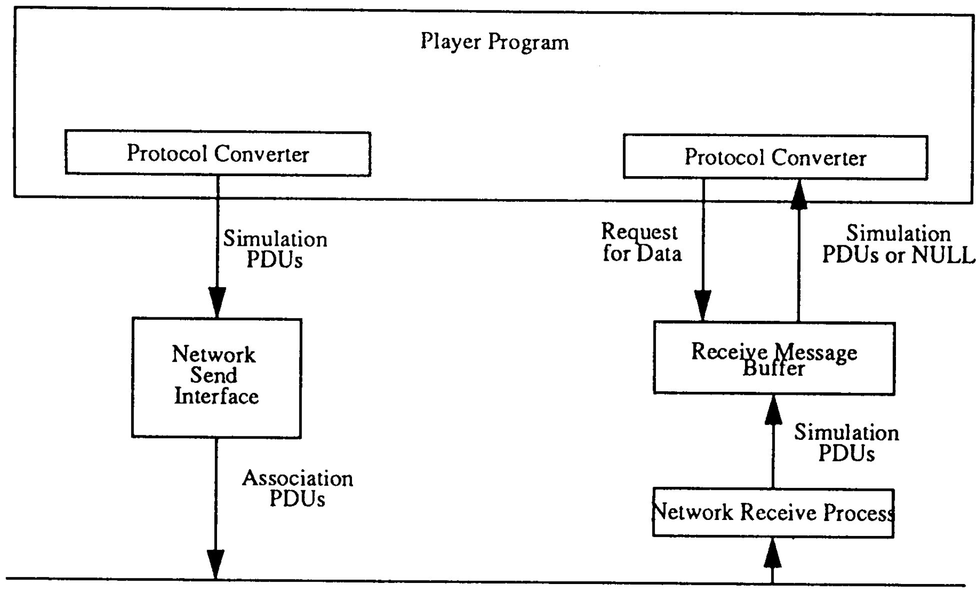

Figure 1. NPSNET Networking Software Architecture

overflowed and crashed. The Warbreaker test was slowed down to accommodate this: only 100 entities were used, with only three of them being high performance aircraft flight simulators. Only ten transient munitions entities were active during the course of the exercise. Semi-automated forces (SAFOR) accounted for the other 70 entities on the network. The average network traffic accounted for 10% of the actual bandwidth.

There are some numbers to know to understand this perhaps a bit better when we talk about scalability of our virtual environments. High performance aircraft typically placed some 20 packets per second onto the network. Slow moving vehicles typically placed some 5 packets per second onto the network. All players, as per SIMNET protocol specification, placed at least one packet every 5 seconds onto the network, just to report in that the player was still alive. Dead reckoning algorithms were are used to minimize packets placed onto the network.

Future Networking & Systems Work

The current NPSNET system is capable of supporting some 500 players using Ethernet and T1 link technology. Like SIMNET, the software architecture of NPSNET is such that all players are recorded in the memories of all workstations on the network. As we begin to study how one handles 10,000 players, we must become smarter and rethink the software architecture from both the networking and systems sides. This systems issue is a strong current focus, with considerations being given to the entire gamut of predicted new computer architectures and computer networks.

Hyper-NPSNET

Hyper-NPSNET is a real time interactive virtual environment with embedded multimedia capabilities. Within the system, there is the notion of an information anchor. These anchors are repositories for a variety of multimedia information. Hyper-NPSNET can have a large number of anchors each with hooks into video, audio, textual and graphics media. Audio in Hyper-NPSNET is stored as AIFC files. Video is stored as SGI Moviemaker files. The user navigates through the system and chooses either directly, or through proximity to an anchor, the multimedia information to view or hear.

In the current system, the user interface consists of multiple Motif panels to administer the information anchors and a SGI GLX Widget for rendering the virtual world. A 3D terrain database is read and used with texture information to create the ground of the virtual world. A multitude of buildings, trees, rocks, telephone poles and other miscellaneous objects populate the world. The end user typically loads an anchor database through the use of pull-down menus and pop-up windows. This database is created through the "Authoring" capabilities of the system (described below). Typically the user chooses to have the anchors visible at all times. This is a visual cue of where the information anchors are attached to the world. The user can trigger any of the available multimedia information by first selecting an anchor and then pressing one of the four buttons: audio, video, graphics or text on the main panel. To select an anchor, the user either selects it with the mouse directly in the 3D world, or chooses it off a list of available anchors displayed in the main control panel of the interface. Upon selection of an anchor, the main panel displays the current anchor name, type and coordinates. In addition, the current anchor is highlighted on the scrolling list of all available anchors. If the user selects an anchor off of the list, then the viewing point in the rendering window is transported to the location of the anchor in the 3D world. This is referred to as an instant aspect change. No matter how the anchor is selected, the user knows what kind of multimedia information is available for this anchor by noticing which of the audio, video, graphics and text buttons are sensitive.

Hyper-NPSNET Navigation

To gain access to all the anchors in the system, the user navigates through the world using either a Spaceball, Ascension Bird or standard 2D mouse. Through trials, it was determined that the most intuitive device and a device found on virtually every workstation was the 2D mouse. The Spaceball made it easy to move around, but difficult to pick anchors within the 3D world. The Ascension Bird had a disconcerting shakiness to the display that could not be overcome.

The user has a number of preferences that can be set through the use of the "Preferences" pop-up panel. Here the user can specify whether to fly around the world or drive on the terrain. If driving, the user steers left or right by moving the cursor left or right outside a small control square in the middle of the rendering window. To speed up, the user moves the mouse up, to slow down or go backwards, the user moves the mouse down. If flying, the heading is set with the left-right motion of the mouse, and the pitch is set using the up-down motion of the mouse. This leaves the speed to be controlled by some other means. The user can specify the flight speed in the panel. The user can also specify whether local anchors only are to be displayed. If local anchors are only being displayed, the user can specify the range that defines what local is. The default value is 300 meters (the current terrain is 2 km by 2 km) meaning that only anchors within 300 meters of the current position of the user are displayed. The last thing the user can specify in the preferences pop-up is whether anchor information is displayed automatically as the user gets close to one of the anchors. If chosen, the user can specify the range used to trigger the multimedia information and can choose what information is automatically displayed. The default is Anchor Auto View off with a range of 20 meters and Audio media tagged. So if the user sets Anchor Auto View on and leaves other default values alone, then anchor audio tracks will play anytime the user gets within 20 meters of an anchor. This is known as *Audio Landmines.*

Hyper-NPSNET Authoring

Hyper-NPSNET can be used as an authoring tool for hypermedia. To build a new hypermedia database, the system is brought up without loading any anchor database. The user then creates all the anchors and attaches all the multimedia using the Anchor Editor Tool. For existing anchors, the editor is used to change any of the values for the anchor name, type, coordinates, orientation, audio track filename, video filename, graphics filename and text filename. For new anchors, the user simply enters all the new information about the new anchor and saves it to the system. An anchor will appear in the 3D location specified by the coordinates and will have all of the multimedia information available for viewing or hearing. Note that the audio, video, graphics and text files have already been created so the role of Hyper-NPSNET is a multimedia player not recorder at the moment. The integration of a real-time audio and video capture capability is a planned and not difficult extension.

Hyper-NPSNET Future Directions

Future directions for Hyper-NPSNET include allowing other than terrain (i.e. fixed) anchors. Useful ones that come to mind include *vehicle* anchors that are attached to moving vehicles. The user could query the vehicle for multimedia information about its design or capabilities. *Temporal* anchors may also be quite useful. Temporal anchors would only exist over some time range and would contain information relevant to the time associated with the anchor. Such anchors would allow the visibility of attached information only during the window of time specified with the anchor. For example, a user of Hyper-NPSNET may only wish to view the video collected in March rather than have the display cluttered with the rest of the year's information anchors.

The system would also benefit from an advanced database system to manage all the hypermedia. This would be very useful for authoring and as an example, would allow the author user to browse all available video clips that match some wild card search parameters. Another possibility with such a system is the ability to visit all anchors that refer to a particular audio track.

Conclusion

A fully interactive version of NPSNET is a continuing developmental effort. We are a long way from being done. At times there are several versions of NPSNET up and running, all prototypes testing out some new capability or feature. We have illustrated the networking and hypermedia software we are working on for NPSNET-4 and Hyper-NPSNET as paradigmatic of the software necessary for the development of future virtual worlds.

Acknowledgments

We are grateful to our sponsors who let us do whatever we want in exchange for unlimited demos. In particular, we thank ARPA/ASTO, the Defense Modeling and Simulation Office, US Army STRICOM, US Army TRAC-Monterey, and the Naval Postgraduate School Direct Funding Program for providing us funds with which to populate the NPS Graphics and Video Laboratory with the tools we need to carry out our "work"!

Biographies

Michael Zyda's main focus in research is in the area of computer graphics, specifically the development of large-scale, networked 3D virtual environments and visual simulation systems. That research is carried out in the Naval Postgraduate School's Department of Computer Science. He is one of the principal investigators of the NPSNET Research Group. Michael Zyda is a Professor in the Department of Computer Science. He is also the Academic Associate and Associate Chair for Academic Affairs in that department. He has been at NPS since February of 1984. Professor Zyda is the chair of the Computer Generation Technology Group of the National Academy of Sciences' Committee on Virtual Reality Research and Development. Professor Zyda was the chair of the 1990 Symposium on Interactive 3D Graphics held at Snowbird, Utah. Professor Zyda began his career in Computer Graphics in 1973 as part of an undergraduate research group, the Senses Bureau at the University of California, San Diego. Professor Zyda received a BA in Bioengineering from the University of California, San Diego in La Jolla in 1976, an MS in Computer Science/Neurocybernetics from the University of Massachusetts, Amherst in 1978 and a DSc in Computer Science from Washington University, St. Louis, Missouri in 1984.

Chuck Lombardo is a member of the research and development staff of the Computer Science Department at the Naval Postgraduate School. Chuck has a BS in Physics and Mathematics from California State University at Sacramento and an MS in Physical Oceanography from the University of Washington, Seattle. He is currently completing an MS in Computer Science at the Naval Postgraduate School where Hyper-NPSNET is the topic of his thesis.

Dr. David R. Pratt is an Adjunct Instructor in the Department of Computer Science at the Naval Postgraduate School. Dr. Pratt is Co-Principal Investigator of the NPSNET Research Group, with Professor Zyda. Dr. Pratt received his Ph.D. in Computer Science from the Naval Postgraduate School in June 1993. He has a BS in Electrical Engineering from Duke University, and an MS in Computer Science from the Naval Postgraduate School. He also has 14 years of computer experience, with 6 1/2 of those years in the Marine Corps.

References

[1] Gelernter, David "Mirror Worlds: or the Day Software Puts the Universe in a Shoebox ... How It Will Happen and What It Will Mean," Oxford University Press, 1991.

[2] Institute for Simulation and Training, "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation", Military Standard (DRAFT), IST-PD-90-2, Orlando, FL, September 1991.

[3] Pope, Arthur, "The SIMNET Network and Protocols", BBN Report No. 7102, BBN Systems and Technologies, Cambridge, MA, July 1989.

[4] Thorpe, Jack "The New Technology of Large Scale Simulator Networking: Implications for Mastering the Art of Warfighting," *Proceedings of the Ninth Interservice Industry Training Systems Conference*, November 1987.

[5] Zyda, Michael J., Pratt, David. R. and Kelleher, Kristen M. "The Software Required for the Computer Generation of Virtual Environments," accepted for a future issue of *Presence*.

[6] Zyda, Michael J., Pratt, David R., Monahan, James G. and Wilson, Kalin P. "NPSNET: Constructing a 3D Virtual World," in *Computer Graphics, Special Issue on the 1992 Symposium on Interactive 3D Graphics*, MIT Media Laboratory, 29 March - 1 April 1992, pp. 147-156.