Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

VIRTUAL REALITY AND SIMULATION: AN OVERVIEW

Robert Macredie, Simon J.E. Taylor, Xiaoning Yu and Richard Keeble

Department of Computer Science and Information Systems
Brunel University
Uxbridge, UB8 3PH, ENGLAND

ABSTRACT

Virtual environments can be used as an effective tool for training and education. This paper cites many recent examples of such systems and divides current work into virtual environment development and the development of supporting technology for distributed access. A novel supporting technology for distribution is introduced.

1 INTRODUCTION

A survey of the recent literature indicates that current research focusing on training is concerned with the development of virtual environments and the development of supporting technology to distribute access. Examples concerning the development of virtual environments show the wide range of applications used for training purposes.

Suzuki et al (1995) describe a driving simulator which allows a user to drive through a virtual town. ROADNET (Katz 1995) is a proposed system which utilises distributed interactive simulation technology to support a network of vehicle simulators. The paper describes the architecture and functionality of ROADNET. Similarly Stevens and Neal (1995) propose the use of distributed interactive simulation to support air traffic control simulation with the National Simulation Capability programme of the Federal Aviation Administration of the USA.

Mizumaki et al (1995) discuss an indoor distribution simulation system based on virtual reality technology. The system is designed to give the user the feeling of touring a substation for routine inspection. It is proposed that the validation of equipment operating procedures, training of unskilled workers and work schedule preparation can be improved by the use of this system.

Merril, Raju and Merril (1995) present a prototype virtual reality surgical simulator for laser iridotomy training. The aim of the work is to help physicians acquire the necessary skills without patient risk. Delingette et al (1995) present a craniofacial surgery simulation testbed which makes extensive use of virtual reality techniques. Virtual reality has been used to train surgeons in new procedures and can determine their competence level before going 'live' (Ota et al 1995). Zeigler et al (1995a and 1995b) discuss a virtual reality medical training system for arthroscopy based on computer graphics and virtual reality techniques.

Coleman (1995) presents the view that leaders in the medical profession suggest that accreditation as a trained physician in the field of endosurgery must begin with a training period in a simulated field. The aim is to develop manual dexterity skills before participating as a clinical assistant to an experienced operator. The Virtual Clinic is introduced as a highly realistic and interactive training system which offers surgeons the chance of gaining clinical expertise before encountering an actual surgical event.

As summarised by Zeltzer, Pioch and Aviles (1995) the US Navy sponsors the Virtual Environment Technology for Training programme which addresses key aspects of this technology. The initial prototype used for evaluation is based on training the officer of the deck on a submarine.

Vegas is a virtual reality application dedicated to the experimentation with evacuation scenarios (Sims 1995). The main goal is to communicate understanding as to the process of a disaster by the use of visualisation to present data generated by the simulator. It is hoped that it will allow users to understand fire and smoke spread. The system has been used to predict egress times from an underground subway station and evacuation times from a cross-channel ferry, as well as from several buildings. Bethel, Jacobsen and Holland (1995) discuss the combination of simulation with virtual reality and visualisation tools to aid training in environmental remediation. Stansfield et al (1995) focus on virtual reality to train people for small team, close quarters operations.

Stansfield et al (1995) present research aimed at developing a distributed, shared virtual reality simulation. Situational training of inspectors and escorts under programs to verify compliance with nuclear non-proliferation treaties is being used to investigate the efficacy of this technique.

Hollands and Mort (1995) discuss the use of virtual reality to visualise the simulation of manufacturing systems. It is observed that people unfamiliar with the simulated processes quickly understand the dynamics and the logic involved, and those familiar with the real system can immediately relate to the simulation output. The authors indicate that it seems probable that many manufacturing systems will not be suitable for, or benefit from, modelling using virtual reality techniques. However, with the provision of affordable virtual reality tools it is hoped that experience will show if the general manufacturing simulation community will find such features worth the extra modelling effort.

Bienvenue, Curtis and Thakkar (1995) investigate if educational researchers can use virtual environments in the classrooms of the USA for training and learning, and if teachers can look at virtual reality as being a discovery tool rather than a means to demonstrate principles.

Wakefield and O'Brien (1995) discuss the creation of a set of virtual reality and simulation tools for the detailed planning of construction activities.

Bowen-Loftin (1995) reports on the use of virtual environments for the training of NASA astronauts and ground-based personnel for a variety of activities such as repairing the Hubble Space Telescope. It is argued that this technology offers significant cost savings and increased training throughput. The move towards a shared virtual environment is discussed.

Macedomia (1995) describe a network software architecture for solving the problem of scaling very large distributed simulations. Logical partitioning of the virtual environment is accomplished via associated spatial, temporal and functional characteristics of entity classes. A vehicle simulation is given as an example of a practical application. Pullen (1995a and 1995b) discusses Distributed Virtual Simulation, a new technology that extends virtual reality to the networked environment. Future requirements of networking resources are discussed to supporting this proposed system. Faigle et al (1995) discuss how high performance computing can further enhance virtual environments.

The complexity and risks associated with the operation of the power transmission and distribution systems have increased the importance of personnel training for such systems. Garant et al (1995) present a virtual reality training simulator prototype to support this activity. The simulator is implemented on the basis of a distributed client-server architecture.

Mastaglio and Callahan (1995) present a virtual environment called the Close Combat Tactical Trainer (CCTT) which is claimed as the first fully distributed interactive simulation compliant training system.

On-going work at Brunel University is currently developing underlying technology to support distributed virtual environments. In this paper we present a novel approach to supporting distribution.

2 TOWARDS A DESKTOP VIRTUAL ENVIRONMENT

Typically, current systems are designed based on the following basic design principles, including:

· there is no central computer for event scheduling or conflict resolution,

· autonomous simulation nodes (computers) are responsible for maintaining the state of one or more simulation entities,

· there is a standard protocol for communicating "ground truth" data; receiving nodes are responsible for determining what is perceived, and

· simulation nodes communicate primarily changes of their state. Dead reckoning is used to reduce communications processing.

A system designed on the above principles suffers from high communication volume.

Object oriented programming is known to be a programming paradigm which deals with complex systems in such a way that maintainability, extensibility and reusability are accommodated (Maffeis 1993). Object orientation provides a natural basis for concurrency since the independency required for concurrent execution is made possible as each object contains its own data and procedures. Inter-object co-ordination is achieved by message passing.

One approach to addressing the problems mentioned above is to propose a new architecture which takes advantages of the features of the object oriented programming in the hope of reducing the volume of messages. In this paper, we present a prototype architecture with an object oriented structure.

This prototype object oriented architecture is composed of non-autonomous simulation nodes. With an appropriate user interface, the nodes appear as an integrated virtual environment to the user. Simulation entities are modelled as simulation entity objects which are distributed at different nodes and are migrated dynamically between the nodes during the simulation run. This mechanism of object management presents the possibility of reducing message volume. Communication efficiency can therefore be improved by two measures. One is to dynamically group closely related simulation entity objects to a specific node. Alternatively, a point to point communication approach can be used instead of broadcast.

The system is currently implemented on a network of IBM compatible PCs, with facilities providing real time communication, an interactive display and interface media.

3 THE SYSTEM ARCHITECTURE

As an object oriented simulation system, the basic system elements are objects located at different machines. These machines are called nodes and are connected by the network. There are two types of objects in our prototype. One class of object is used to manage system facilities such as keyboard and mouse operation, network operation and display operation. These objects are usually supplied by system support packages. The other class of object is used specifically for simulation. In this paper we focus on the class of simulation objects.

At any single node, there are three major objects: the message handling object, the situation display object and the simulation object.

3.1 The Message Handling Object

The message handling object controls the messages passed between local objects and the network. Unlike current approaches which broadcast all messages on the network (Pullen, 1995), it uses both broadcast and point to point approaches to send the messages to keep the number of redundant messages as low as possible. It broadcasts all messages about the updated states of local entity objects to other nodes via the network. Control is achieved by a point to point communication approach. The message handler also receives the messages from outside the node and distributes them to the proper local objects.

3.2 The Situation Display Object

With update messages, the situation display object is used to display the states of all entity objects distributed at the nodes. A function may be used in this object to restrict some object state display according to the different training scenarios. The display updates with simulation time advance. The update messages are from the local simulation object and the message handling object. The messages from the local simulation object include all changes of its derivative entity objects' states. The messages from the message handling object are the update messages from all other entity objects located at the other nodes. On some occasions "Dead Reckoning" algorithms can be used in this object to reduce the transmission of state updates.

3.3 The Simulation Object

The simulation object at a node contains a series of entity objects which represent physical entities in the real world (a simulation object controls its derivative entity objects simulation in each time loop). It is responsible for creating and deleting its derivative entity objects. It transfers the messages from its derivative entity objects to its other derivative entity objects or to the other objects located at the other nodes via the computer network. Another important task of the simulation object is to control the local simulation time advance. More detail about entity object management and simulation time management will be introduced in the rest of the paper.

4 SYSTEM MANAGEMENT

4.1 Object management

1) Entity object

In our prototype of system, entities in the real world are modelled by entity objects which are located at different nodes in the system. Each entity object has a unique identity that distinguishes it from other entity objects. During each simulation time loop, each entity object is activated by different messages and the entity object states can be updated. At each node, there is a table recording distribution of all entity objects in the system. When an entity object is created, deleted or transferred to an other node, a message is sent to all nodes in the system to inform them of this change and update the entity objects' distribution tables.

There is an owner relation between the entity objects. When we say the entity object A is the owner of the entity object B, it means the entity object A controls the entity object B in them system. The ownership can be changed. Each entity object has at most one owner. The ownership is used to judge if the entity object should be migrated or not. To reduce control messages passed over the network, any entity object should be located at the same node as its owner object. An attribute in the object is used to describe the ownership between entity objects.

2) Object list

At each node, a list is used to link all entity objects located at local nodes. During each simulation time loop, all entity objects are activated in the list order by control messages which are transferred from the simulation object. With the entity objects location changing, the list is dynamically changed by the system. When a new entity object is created or moved to a node from another node, it is automatically added at the end of the list as a result of its constructor. When an entity object in the list is deleted or moved out to another node, it also automatically removes itself from the list using itsdestructor.

3) Object Migration

Reducing the messages passed via the network and keeping the system load balanced can improve the system efficiency. This can be realised by object migration. Here we only discuss entity object migration.

One of the basic functions of entity object migration is to group entity objects together which have common ownership. Control messages are passed between entity objects. These messages are used by one entity object to control another entity object. They are only passed between the entity objects which have common ownership. Since messages passed locally are more efficient than the messages passed via the network, keeping control local can improve system efficiency.

Ownership can be changed during the simulation. When the owner of one entity object is changed, the entity object is immediately moved to the node where the new owner object is located.

Object migration also facilitates load balancing. Entity objects can be created, deleted, or moved dynamically during the simulation session. At each node, there is an entity object list recording the entity objects located at that node and an entity object location table recording the location of other entity objects at other nodes. A detection object is used at each node based on the above list and table to detect the load states of nodes in the system. When the load of one node is heavy while there is a node with no entity object located, some entity objects can be moved to the idle node to maintain load balance.

4.2 Time Management

Time management is used to control the advancement of simulation time during the simulation session. The simulation object at each node controls time management. To control the simulation time advance, three different time notions should be distinguished in time management.

Tli refers to the earliest message timestamp in the *i*th entity object. Tl refers to the earliest message timestamp in all entity objects. So

$$Tl = \min \{ Tli \mid i=1,...n \}$$

Here *n* is the number of entity objects in the simulation.

Wallclock time Tw is a scaled real time. A scale factor *f* is used to compress or expand real time. It is variable. Different scenarios can set different values of scale factor *f*. So

$$Tw = f * Tr$$

where Tr is the current real time

Simulation current time Tc is used for each simulation loop time advance.

$$Tc = \min \{Tl, Tw\}$$

When Tc = Tl, the entity objects process their own messages with time stamp Tl and change the related states. The other objects only change states which are related to time. When Tc = Tw, all objects only update the states which are related to time. Usually, the logical time is earlier than the wallclock time, but sometimes, there could be delay caused by nodes processing or network communication processing. So an adjusting value Ta should be added to the wallclock, i.e.

$$Ta = Tdelay+Tprocess,$$

where Tdelay is message delayed time comparing with its timestamp, Tprocess is the time the message object needs to process the message. So, the adjusted wallclock time is

$$Tw' = Tw+Ta$$

At each node, there is a message list in timestamp order. Each message is processed when the timestamp is the same as simulation time.

4.3 Message Management

Similar to the classes mentioned above, there are two major kinds of messages in the system. One kind of message serves the operating system, i.e. opening or closing a window, expending or reducing the size of window, moving the mouse, etc. The other kind of message is used for simulation execution, i.e. entity objects state update, an event happening, moving an entity object, etc.

For efficient communication, a broadcast approach is used for entity object state update and a point to point communication approach is used to transmit control messages between entity objects which are distributed at different nodes.

The simulation object at each node distinguishes which messages are only used for local entity objects and which messages should be transferred to the message handling object.

Message order affects the whole simulation event order, especially concurrent events. All messages have a time stamp marking effective time. All entity objects will process the messages in time stamp order. This guarantees the processing of concurrent events in the correct order. All message handling objects in the system run in parallel and are synchronised. In each time loop, message handling objects send all messages created by local entity objects to other nodes. When a node has no messages to broadcast, it broadcasts a time advance request message to inform other nodes that it has nothing to send and requests time advance. At the same time, each message handling object holds all messages until it has received messages from all nodes and then transfers them to the display object and simulation object in timestamp order.

5 Conclusion

This paper has introduced virtual reality and simulation within a training context as being a field which concerns interface design and distributed access. We have described a novel architecture which will be used to underpin future in virtual environments.

REFERENCES

Bethel, W., Jacobsen, J. and Holland, P. 1994. Site remediation in a virtual environment. Proceedings of the SPIE, The International Society for Optical Engineering, 2178: 78-87.

Bienvenue, L. A., Curtis, D. H. and Thakkar, U. 1995. Virtual environments in K-12 learning and discovery: a grand challenge in education? Computer Graphics, 29(4): 43-4.

Bowen-Loftin, R. 1994. Virtual Environments for Aerospace Training. In Proceedings of WESCON/94, Idea/Microelectronics, 384-7. IEEE, New York, USA.

Coleman, J. E. 1994. Virtual reality in medicine. In Proceedings of VR '94, the Fourth Annual Conference on Virtual Reality, 169-73. Mecklermedia, London, UK.

Delingette, H., Subsol, G., Cotin, S. and Pignon, J., 1994. Virtual reality and the simulation of craniofacial surgery. In Proceedings of Montpellier '94. 3rd International Conference. Interface to Real and Virtual Worlds, 399-408. EC2, Nanterre, France.

Faigle, C., Fox, G. C., Furmanski, W., Niemiec, J. and Simoni, D. A. 1993. Integrating virtual environments with high performance computing. In Proceedings of the IEEE Virtual Reality Annual International Symposium, 62-8. IEEE, New York, USA.

Garant, E., Daigle, A., Desbiens, P., Okapuu-von Veh, A., Rizzi, J.-C., Shaikh, A., Gauthier, R., Malowany, A. S. and Marceau, R. J. 1995. A virtual reality training system for power-utility personnel. In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 296-9. IEEE, New York, USA.

Hollands, R. J. and Mort, N. 1995. Manufacturing systems simulation: mixed mode and virtual reality simulation. In Proceedings of Fourth International Conference on Factory 2000 - Advanced Factory Automation, 651-7. IEE, London, UK.

Katz, W. 1994. ROADNET - distributed interactive simulation applied to driver training, city planning and transportation research. In Proceedings of the Twenty-Sixth Annual Summer Computer Simulation Conference, 936-41. SCS, San Diego, California, USA.

Macedomia, M. R., Zyda, M. J., Pratt, D. R., Brutzman, D. P. and Barham, P. T. 1995. Exploiting reality with multicast groups: a network architecture for large-scale virtual environments. In Proceedings of the Virtual Reality Annual International Symposium '95, 2-10. IEEE Computer Society Press, Los Alamitos, California, USA.

Maffeis, S. 1993. "ELECTRA - Making Distributed Programs Object-Oriented".: Proceedings of the Symposium on Experiences with Distributed and Multiprocessor System IV, USENIX. San Diego, California, USA.

Mastaglio, T. W. and Callahan, R. 1995. A large-scale complex virtual environment for team training. Computer, 28(7): 49-56.

Merril, G., Raju, R. and Merril, J. 1995. Changing the focus of surgical training. VR World, 3(2): 56-8, 60-1.

Mizumaki, Y., Morita, S., Asano, K. and Kamiji, N. 1995. Power substation simulation system using virtual reality technique. Transactions of the Institute of Electrical Engineers of Japan, 115-C(2): 267-72.

Ota, D., Loftin, B., Saito, T., Lea, R. and Keller, J. 1995. Virtual reality in surgical education. Computers in Biology and Medicine, 25(2): 123-37.

Pullen, J.M. and D. C. Wood. 1995. "Networking Technology and DIS", Proceedings of the IEEE, 83(8): 1156-1167.

Pullen, J. M. 1994a. Networking for distributed virtual simulation. In Proceedings of INET'94, the Annual Conference of the Internet Society, held in conjunction with JENC5, the 5th Joint European Networking Conference, 241-7. Plattner, B. R. and Kiers, J. P. A. The Internet Society, Reston, Virginia, USA.

Pullen, J. M. 1994b. Networking for distributed virtual simulation. Computer Networks and ISDN Systems, 27(3): 387-94.

Sims, D. 1995. See how they run: modeling evacuations in VR. IEEE Computer Graphics and Applications, 15(2): 11-3.

Smith, A. B. 1995. Using virtual reality for the simulation of infrared environments for human training. In Proceedings of the 33rd Annual Southeast Conference, 101-9. ACM, New York, USA.

Stansfield, S., Shawver, D., Miner, N. and Rogers, D. 1995. An application of shared virtual reality to situational training. In Proceedings of the Virtual Reality Annual International Symposium '95, 156-61. IEEE Computer Society Press, Los Alamitos, California, USA.

Stansfield, S., Shawver, D., Rogers, D. and Hightower, R. 1995. Mission visualisation for planning and training. IEEE Computer Graphics and Applications, 15(5): 12-14.

Stevens, R. J. and Neal, W. J. 1994. Distributed air traffic management simulation: the next application-domain for DIS. In Proceedings of the Twenty-Sixth Annual Summer Computer Simulation Conference, 828-33. SCS, San Diego, California, USA.

Suzuki, T., Yasuda, T., Yokoi, S. and Toriwaki, J. 1994. Construction of virtual town and driving simulation in it using graphics workstation. Journal of the Institute of Television Engineers of Japan, 48(10): 1318-25.

Wakefield, R. R. and O'Brien, J. B. 1994. A 'virtual reality' type simulation system for construction automation system development. In Proceedings of Automation and Robotics in Construction XI, the 11th International Symposium on Automation and Robotics in Construction, 239-47. Chamberlain, D. A. Elsevier, Amsterdam, Netherlands.

Zeltzer, D., Pioch, N. J. and Aviles, W. A. 1995. Training the officer of the deck. IEEE Computer Graphics and Applications, 15(6): 6-9.

Ziegler, R., Fischer, G., Muller, W. and Gobel, M. 1995. Virtual reality arthroscopy training simulator. Computers in Biology and Medicine, 25(2): 193-203.