3D Training Environments

VRML and its use in Interactive Task-based Simulations

Seamus Brady
CBT Systems, Dublin

Carol O'Sullivan
Image Synthesis Group, Trinity College Dublin

Abstract

Training in the IT sector is vital. With technologies evolving ever faster, people are required to learn new skills all the time. Instructor-Led Training (ILT) is popular and a useful method of training but is expensive and companies are looking towards Computer-Based Training (CBT) to provide cheaper methods of educating the work force. There are already many CBT applications in the marketplace, all of which use 2D interfaces. This paper examines the necessary requirements for a 3D interactive training application, and looks at the technologies that exist to support and implement such an application. The goal is to build a prototype interactive training application that allows users train with 3D models and have the application monitor and guide the user through training tasks.

1 Introduction

Applications and technology are evolving rapidly and employers are finding it necessary to spend more time and money training their employees. Computer-based training (CBT) is gaining widespread acceptance in the marketplace as a cheap supplement or even alternative to Instructor-led training (ILT). Virtual Reality (VR) technology is being used for training in many different sectors but because of the cost of implementing the technology, it is still inaccessible to the marketplace at large. The Virtual Reality Modeling Language (VRML) is a technology that can bring VR over the Internet and onto the desktop and make 3D much more accessible to the end-user. An Internet-based training application has been created, that can provide task-based training in a Virtual Environment (VE) while also taking into consideration the latest training methodologies and interface design issues.

1.1 Task-based Simulations

A task-based simulation in the context of Computer-based Training (CBT) is an application that simulates or replicates a user-task in the real world [CBT, 1995]. Generally, tasks are confined to simulating user-tasks in computer applications. For example, the application could simulate the look and feel of Microsoft Access and the particular task could be to create a new database. Tasks involve a number of steps, some of which may have multiple correct answers. It should be possible for the user to complete the task in the application whatever way they choose and they should not be able to easily distinguish between the simulation and the real application within the confines of the task at hand. Obviously, it is not possible to completely simulate the whole application but the goal is to allow the user as much freedom as possible so they are not shepherded towards the answer.

1.2 Defining the Interface

To define an interface for 3D task based simulations, there are two main considerations

- provide interface components to easily allow the user to manipulate the 3D environment
- provide interface components to allow the user to receive information about the task at hand

In the first instance, the user will be working with a 3D environment. They will require the ability to manipulate 3D objects. In many cases the user will also require the ability to navigate within the environment, i.e. they will need to be able to move around objects and be able to change their perspective of the environment.

Secondly, the user will be interacting with the 3D model by executing some task. The information on the task and steps needs to be presented. Also, there should be a method of allowing the user to inform the application when they have finished working with the 3D model. Then the application needs to present feedback on the task and lastly, they need the ability to tell the application that they would like to see a demonstration of the task.

These requirements lead to two different types of interface components: one that provides the 3D functionality necessary and one that provides the more straight-forward 2D functionality.

1.3 VRML

The Virtual Reality Modelling Language (VRML) is a file format for describing interactive 3D objects and worlds. VRML is designed to be used on the Internet, Intranets, and local client systems. VRML is also intended to be a universal interchange format for integrated 3D graphics and multimedia. VRML may be used in a variety of application areas such as engineering and scientific visualisation, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds.

2 Research

2.1 Virtual Reality

'Virtual Reality' (VR), initially coined by Jason Lanier [Lanier, 1989], is used by many different people with many different meanings. There are some people to whom VR has a very specific meaning and who think about Head Mounted Displays (HMD), Glove Input Devices and Immersive systems. Some other people stretch the term to include movies, books and even pure fantasy and imagination [Isdale, 1998]. To help clear up the confusion about what VR actually is, Professor R.S. Kalawsky defined a generic model of a VR system [Kalawsky, 1996]. Essentially, when a person interacts with a Virtual Environment (VE), they become engaged in a closed loop. The user interacts with the VE representation, which in turn is based on a model. The effects of this interaction are fed back to the user through the modification of the model based on the user's input and so the cycle begins again. Lanier himself describes Virtual Reality as an immersive, interactive simulation of realistic or imaginary environments.

2.1.1 Types of VR

There are three basic varieties of VR systems: non-immersive (desktop) VR, semi-immersive (projected) VR and fully-immersive VR.

Non-immersive VR is the most common and inexpensive type of VR. In its most basic form it consists of little more than a computer generated VE and conventional input devices. There is no sense of immersion – or being in the VE.

Semi-immersive VR is characterised by a fixed wide-angle display of over 60 degrees in diameter. The most common type is projected VR where multiple users gather in a room resembling a small film theatre while watching a VR display on a large screen that curves towards them at the sides giving them a wide angle view of about 130 degrees.

Fully-immersive VR is the form of VR that most people imagine when talking about Virtual Reality. It consists of a head coupled Visual Display Unit. Any movement of the users head is fed into the system and the users view is updated accordingly. This gives the user the feeling of being immersed in the system and there is no visual sense of the physical world or even of a computer interface [M. Bricken, 1991]. Fully-immersive systems generally include haptic devices which allow the user to feel simulated 3D objects in the VE. The most common type of haptic device is the "cyberglove". Other haptic devices include the exoskeleton and the thermode (which provides the sensation of temperature).

2.1.2 Attributes of VR

VR has been categorised by some people as just an extension of Multimedia systems [Dede, 1992]. In an effort to empirically measure VR in relation to other computer-based systems, [Zelter 1992] has defined a framework based on 3 independent variables: Autonomy, Presence and Interaction.

Autonomy reflects the extent to which an environment operates on its own - without requiring user input. Presence reflects the extent to which the user feels immersed in the system.Interaction reflects the systems responsiveness to the users actions.

Each system can be categorised as having various levels of autonomy, presence and interaction. For example, TV is high on autonomy and low on interaction, with presence being variable depending on the user and the program. Computer tutorials are low on autonomy and presence but high on interaction.

VR can be high on all three, depending on how immersive it is. Desktop VR provides the lowest level of presence. Depending on the VE, the level of autonomy can be high. For example a real-time VR simulation is highly autonomous.

It has been shown that varying these three parameters in a learning technology has a profound effect on the amount of learning performed. Interaction must remain high. This helps to keep students focused and motivated. If presence is high, users are less likely to be distracted by outside circumstances. High autonomy is advantageous in learning environments where time is an important factor.

2.2 Theory of Learning

There has been a change in the under-lying theory of how people learn by use of technology-based educational systems. It is due to this shift in thinking, that VR has found uses in education. In fact, educational computing has gone through three major changes and VR has opened up a fourth, called "Constructivism". [Winn, 1993]

Constructivism is based on the belief that the nature of the interaction between the student and instruction is a determinant of learning of equal if not greater importance than content or how information is presented [Merrill, 1993]. Indeed, this theory prescribes that all knowledge is derived through interactions [Spiro et al., 1991]. The basis of this theory is that each individual has his own internal representation of the outside world and when he interacts with the world, he amends his internal structure of the world. Because each individual is unique, people cannot share "first-person" knowledge but only with the use of symbols, exchange "third-person" knowledge. In other words a person cannot know what it feels like to be another person, they can only form "third-person" knowledge of the experience [Winn, 1993]. Two people will build similar internal structures of the world but each will have their unique interpretation of these structures.

The relationship of constructivism to VE is that it provides a theoretical framework for the type of learning that is already taking place in fully immersive VE's. The key to the compatibility of VR with constructivism lies in the notion of immersion. According to constructivist theory, knowledge construction arises from first-person experiences that can never be entirely shared. Immersive VR allows first person experiences by removing the interface between the computer and the user. It also allows the user to learn through interactions with the VE and it is through the interactions that the knowledge is formed.

2.3 Graphical User Interfaces

Each VR system has some sort of control panel or interface for the user to interact with. The interfaces vary widely depending on the type of VR, the input/output devices available and the method in which the interfaces are created. The type of interface used has a significant bearing on the user's perception of the application [Norman, 1986].

There are several ways to create the control panel for a VE using either 2D or 3D controls. It is possible to design an interface as part of the 3D model. Such an interface would float in front of the users field of vision and remain in sight no matter how the user moves around. This is called a Heads-Up Display (HUD). There is not much room on the HUD to display all the information required in a 3D task-based training application – such as the task text and the step details. This point indicates the necessity of using 2D interface components to allow the task and steps information to be easily displayed.

Ideally a 2D interface should be designed such that:

- it is independent of the 3D model or content
- it can be easily replaced without affecting existing content
- it provides the interactivity necessary for communicating with the 3D model

In short, it is better to provide this information outside of a VRML model and use Java or HTML to manage the 2D interface.

2.4 Uses in Training

VR has many applications, the main one being in training and education. VR has been used for many years in the Aeronautical Industry for training pilots and is now an indispensable tool. Pilots would be unable to gain experience in critical situations and the training of new pilots would be a lot more hazardous. VR has also been found useful in the military sector, with virtual combat missions and virtual tactical warfare being used for training.

VR has also found other areas of use. Much research has gone into its use in education and training [e.g. Fox et al., 1994; Youngblut, 1998] Fire fighters have fought virtual blazes, surgeons have training on virtual patients [Warrick, 1996] and Engineers have built virtual machines. VR has many uses for education but how does a person decide when VR is beneficial for use in training.

2.5 Evaluating Knowledge Retention

[Hall et. al, 1998] developed a VR application to evaluate users knowledge retention and compared their results to a 2D version of the application. Users were required to recall a ten-step procedure for operating several devices distributed around a room. The 2D version used graphics and photographs to display the devices. Buttons and windows allowed users to get information on the devices. The 3D version provided a virtual room with the devices distributed throughout the room. Users were required to navigate between the objects and a glove input device was used to manipulate controls on the devices.

Analysis of the performance of groups of users on each application type showed that users of the VR application had marginally higher test scores but the test groups were too small to provide concrete statistical evidence that VR was a better technology for training. Rather, the conclusions were that the instructional design methodology was more important than the actual technology being used. Training programs that encourage the users to learn and think will always out-perform training programs that simply rely on a new technology.

2.5.1 NICE project

The NICE (Narrative-based, Immersive, Constructionist/Collaborative Environments) project [Roussos, 1997; Johnson et al., 1998] was a research project whose goal it was to create a virtual learning environment, based on the constructionist theory. The design of NICE supports the constructionist view that learners assimilate knowledge by engaging in self-directed learning activities which are accomplished through constructive tasks. Users discover relationships between variables through the direct manipulation and examination of objects in the VE.

Users were not presented with tasks or objectives or directed in the environment, which reduced the possibility of them learning. This lack of directness leads to the conclusion that constructionist theory is more successful if users have a goal. Their conclusions also indicate that while VR brings "added value" to training, there is little reason to use VR in areas that are already well met by conventional pedagogy. They also stress that the learning goal of VR applications must be hard. In other words, the objective must be difficult to obtain through the use of existing learning models and the advantages of VR must be utilised in the learning process.

3 Implementation

3.1 Objectives

The overall objective is to build a sample application that demonstrates the use of VRML as a medium for 3D task-based training simulations. Ideally, the application should be reusable. It should be possible to insert various 3D models easily without having to redesign or develop parts of the application. To achieve this, a number of sub-goals must be reached:

- Define an interface for the user to interact with the 3D model
- Build the 3D models and build the interactivity into the models
- Develop a judging mechanism that can be abstracted from the example and applied in general
- Develop a demonstration mechanism that can be abstracted from the example and applied in general
- Integrate the components together

To make the application reusable, it should be possible to insert 3D VRML models that meet some defined specification without having to make any changes to the basic application. The critical components are how the application can communicate with the model to determine the status of objects that pertain to the steps in the task and also how these objects do their own demonstrations under the instruction of the application.

3.1.1 Interface design

Figure 1 shows a screen-grab of the interface that has been implemented for this project. The graphic shows the VRML controls for an "Examine" paradigm. The developer can choose which set of controls to display depending on the VRML content – "Examine" or "Movement". It is also possible to have both sets of controls available to the user.

Figure 2 shows the application structure. The user can interact directly with the VRML model using the VRML interface. For accessing task details such as the steps or feedback and resetting the task or exiting the task, the user interacts with the Java applet. For changing views in the model or activating the demonstration, the user interacts with the HTML pages for the steps and feedback respectively, both of which contain JavaScript functions that send messages to the Java applet. The applet is wired up to the VRML model for communicating requests and for receiving information back.

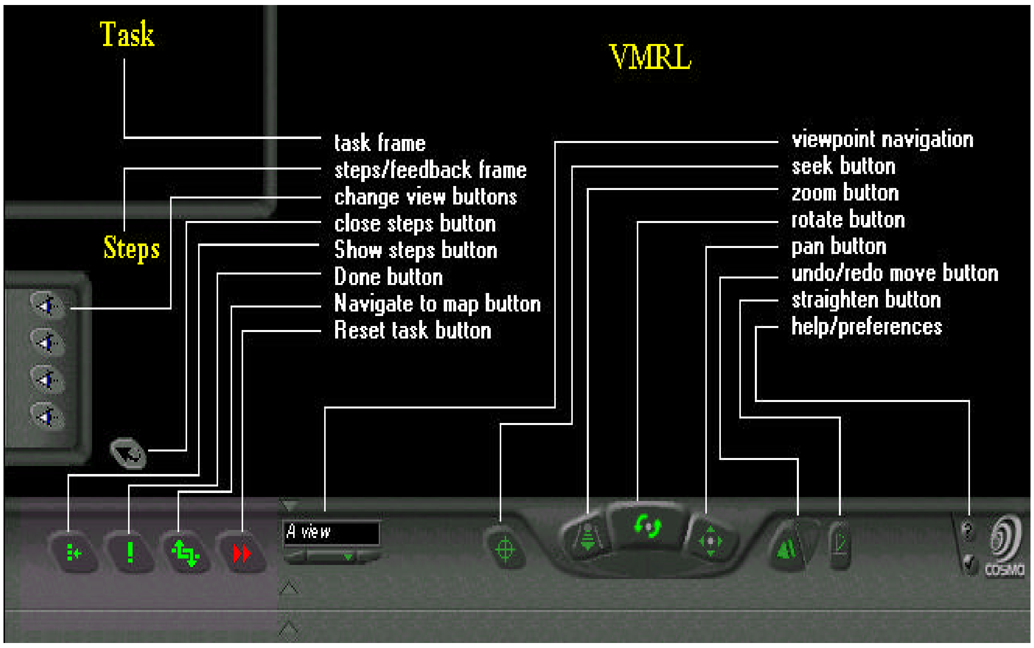


Figure 1 Application interface

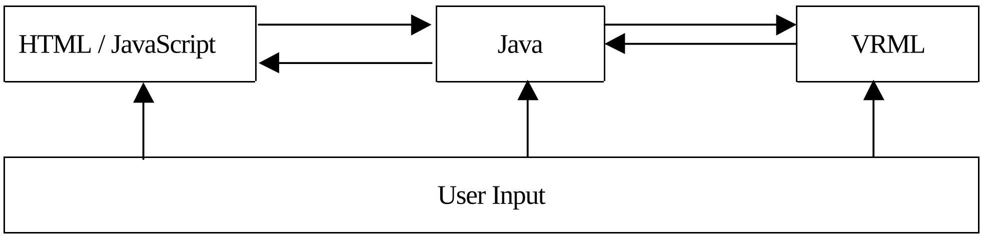


Figure 2 Application structure

3.1.2 Java Task applet

The Java Task applet sits in the task frame and provides the interface for the user to control the task. The following functionality is available:

- Displaying steps for the task
- Judging the user on the task
- Displaying the correct and incorrect feedback
- Resetting the task to the initial state

The applet has been designed so that is a generic component of the application. The developer can change the 3D models and the task without having to rewrite the applet or change it in any way. This was possible because the architecture of the models was defined so that there is always a generic interface for the applet to communicate with. As long as the 3D models meet the specifications, they can be integrated into the application.

3.1.3 3D Model Design

Building 3D task-based simulations means building objects and events that reflect reality, ie. real-world actions. In general these interactions can be broken down into a number of discrete actions. The basic actions needed in an interactive 3D application are:

- Moving viewpoints / changing the view
- Navigating though environment
- Selecting an object
- Moving an object

Examples of each of these interaction types have been developed in the 3D models designed for the application.

3.1.4 3D Model Architecture

Each 3D model integrated into the application has one common feature. That is a component that tracks the internal changes in the model and provides an interface for the Java applet to communicate with. This component is called the "Global Test script node".

When the application applet is first started, it forms two-way links with this component (

Figure 3). Once these links are established, the Java applet can receive messages from and send messages to the VRML model. For example, when the Done button is clicked in the Java applet, a message is sent to the VRML model, requesting the status of the steps. The Global Test script node keeps track of this type of information and sends a message back to the applet. The applet can then determine what feedback to provide the user on the basis of the step states.

For the demonstrations, the applet again sends a message to the Global Test script node, which activates the appropriate demonstration. The demonstrations (which may be animations) are developed as part of the 3D model and are linked into the Global Test script node.

3.2 Building the Models

A number of 3D models have been built to test various aspects of the application. The first model is quite simple and was designed to test the Java applet and the generic components of the application. The second concentrates on the elements of VRML that allow interactivity and demonstrates the types of interactivity that is possible. The third highlights the generic design of the application by showing how a pre-built model can be integrated into the application with the minimum of effort.

3.2.1 Test Model

To test that the architecture for the application was suitable, a simple VRML world was designed to represent an interactive 3D model. The judging model and rules for performing steps were applied to this model. The 3D model used in this example, simply has 4 spheres which can be moved left or right by clicking on them. The task is to move all the spheres to the right hand side. The initial state of each sphere is in the left position. The task is judged to be correct if all spheres are in the right position. The user is able to do and undo each step. However there are restrictions on the order in which the spheres can be moved. These rules are incorporated into the VRML model in the Global Test script node, which manages the steps.

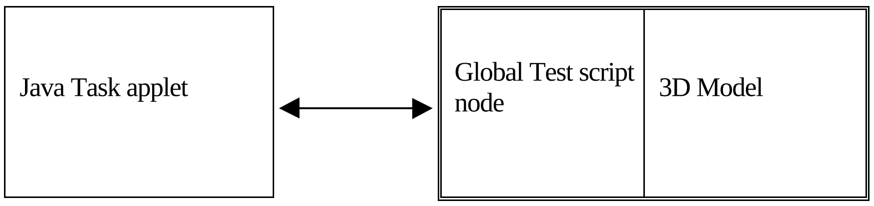


Figure 3 Communication between applet and 3D Model is through the generic Global Test Script node

3.2.2 Office Simulation

The Office simulation was developed to test not just the reusability of the application, but also to test the usability of VRML for building interactive models. Without interactivity, task-based simulations are not practical and an examination of the techniques available in VRML is necessary to test the viability of the language. The simulation consists of a room with two desks and a moveable chair. Each desk has sliding drawers. On one desk is a phone, which can ring, and a standard computer with an adjustable monitor. On the second desk is a laptop. In the simulation, the user is required to find a CD which is hidden in one of the drawers, take it and put it in the laptop CD tray - which must first be opened, and then open the laptop display. The simulation does not focus on a learning task as such, but rather highlights the basic elements that are necessary for the developer to be able to build an interactive simulation. Different sensor types, multiple viewpoints, animated demonstrations are all employed in this task.

The first simulation utilised only touch sensors to activate the steps. The office simulation uses most of the different types of sensors that are available.

3.2.3 Medical Simulation

The Medical simulation focuses on a typical learning task and highlights an area of use for 3D models that is particularly important. It also tests the usability of the application with pre-built 3D models and the use of alternative correct answers.

In this simulation the user is required to identify 4 specific anthropometric points on the human head out of a possible forty. The points are placed all around the head so it is important for the user to be able to move around the head and examine it in detail - some points are very close together. The user must match the four specified points with the panels A, B, C and D. The user is not required to pick the points in order and they can redo any of the steps until they think they have got them all correct or have given up. Some of the steps have more than one correct answer and it doesn't make any difference which one the user picks as both are marked correct. If the user requests a demonstration of any step, the viewpoint is changed so that the anthroprometric point is visible and then the point is enlarged to distinguish it from the other points. It is of course possible to code more effective demonstrations, such as an animation of a 3D hand pointing at the correct point.

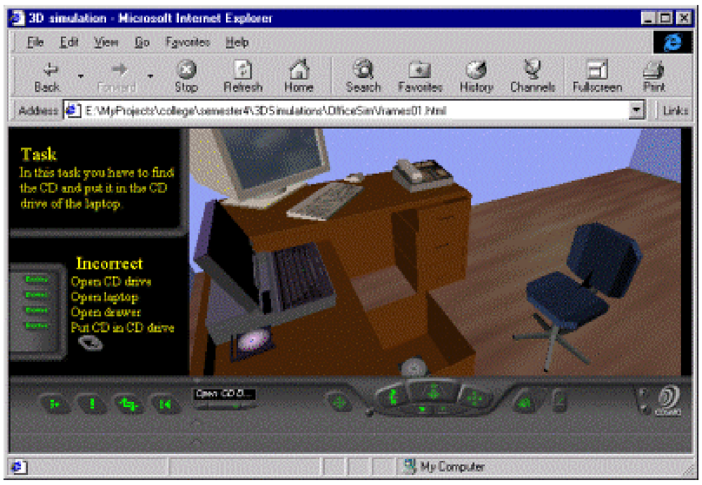
The model of the head itself was developed by Ressler et al. [1996] and then modified and integrated into the simulation. The model initially consisted of just the head and the anthropometric points. To this, the panels were added and of course the Global Test script node. Viewpoints and Demonstrations for each step in the task were also added. Once the model was complete there were over 160 active events to provide the interactivity and monitor the users progress.

3.3 Rules for building the models

- When a 3D model is being built for the application there is no need to worry about how to link the objects together for the different steps. An abstract model has been defined which lets developers build the models in such a way that allows them to be easily copied to different models without worrying about the interdependencies between objects. The general guidelines for building the models are listed here. The sample models should be examined in detail to see the actual implementation.

- Each step of the task should have a viewpoint defined for the object referenced in the task such that the object in the step is visible in both the beginning and end state. The user should not be required to change the view in order to complete the step.
- Each step should have a script node that can trigger the demonstration for the step, know when the user has activated the step and fields for storing the current state of the step.
- The object for each step should be grouped together with the sensor that activates the step and the script that manages the step
- Each model should include the Global Test Script node and link its objects to it.
- Objects should be built with extensive use of the “Transform” node. (The Transform node is a grouping node in VRML that defines a co-ordinate system for its children that is relative to the co-ordinate systems of its ancestors. It also allows the user to build up a hierarchy of objects that makes it much easier to navigate through.)
- Each object used in a step should have a beginning state and an ending state. For example an object can move from A to B or change value from X to Y.
- How an object gets from the beginning state to the End State is left to the developer and depends on the action being performed. In general, there are two ways of arriving at an end state. A) The object can be animated going from the beginning to the end state. B) The object can jump immediately from the beginning state to the end state. In any case there should be some sort of trigger than can be accessed in order to allow the application to interact with the step. This trigger can be a script node which controls the object or can be some event that is implicitly defined in the VRML model.

With just these guidelines a designer can build models independently of the application. However, if these guidelines are not followed, then a 3D model cannot be easily integrated into the application.


3D simulation - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Print
Address
Links
My Computer

Task
In this task you have to find the CD and put it in the CD drive of the laptop.

Incorrect
Open CD drive
Open laptop
Open drawer
Put CD in CD drive

Figure 4 Office Simulation

4 Discussion

The overall goal of the project was to build an application with various 3D models that demonstrate the applicability of VRML to 3D task-based simulations in the context of current learning methodologies. This has been achieved with the development of the first web-based 3D task-based training application. The application design provides a framework that allows different 3D models to be easily integrated into the application to provide task-based training.

Each of the three models developed for the application highlights some of the important aspects of 3D interactive models and their use in training.

- Interface design
- Reuse of application framework
- Incorporation of task-based training recommendations
- Animations
- Different methods of interactivity
- Movement of objects through VRMLs hierarchy
- Mapping real-world user-object interactions to VRML user-object interactions.

4.1 Application Features

The application was designed with the task-based training goals in mind. Each of the recommendations were implemented:

- Task-based judging
- Task statement always visible
- Detailed steps available
- Demonstrations for each step
- Accommodation for multiple correct answers
- Ability to reset task to initial state

Each of these functions could be accessed through the interface. The various components in the interface have a consistent look and feel. The VRML plug-in has been integrated seamlessly into the application. This makes it easier for the user to learn how to use the interface as there is only one style of interface to learn.

4.2 Interface Design

In designing the interface, the style of the VRML interface was used as a basis for the design for the rest of the application. This required the look and feel of the interface elements to be similar to the VRML interface. It is possible to design a totally different interface and replace the existing VRML one, but this requires some extra development work and increase the size of the application to be downloaded.

Localisation issues were considered in the design of the interface. There are no interface graphics with text in them. Each button has a tool tip and a line of help text, which is coded in the applet. It is a straightforward step to localise the interface. Also, the content is very easy to localise because both VRML and HTML are text based and strings can be easily edited without having to make any changes to the application itself.

While the use of viewpoints is important and makes navigation and exploration of a virtual model easier, associating viewpoints with particular steps is somewhat leading. It is possible for the user to guess the correct action for a step if the viewpoint is focused on a particular object or component. It is felt that it would be better to provide a separate interface for the user to change the viewpoints. Indeed, the VRML Browser allows a user to do just that, but that is only a solution if its interface is used. Another alternative is to provide a simple popup list in the Task applet that the user can choose the views from or even to provide a 3D-spinning cube [Cosmo, 1998] where the user can click on a side to bring up a view.

4.2.1 Building the models

Building the 3D objects and components for the model can be time-consuming and use of a GUI based editor is essential for developing the models. Cosmo Worlds, which is probably the best VRML editor available at the moment, was used to develop the models in the application. Even with the use of this tool, it is sometimes necessary to use a text editor to modify the VRML code directly to get the required results. This has implications on the level of IT awareness of the developer.

The models that Cosmo Worlds creates are compatible with Cosmo Player 2.1 which ships as part of Netscape Navigator but which requires a plug-in for Microsoft Explorer. Other VRML plug-ins are available but because of some differences in implementation, some models do not work correctly between different plug-ins. The models developed for the application have only been tested with Cosmo Player 2.1 plug-in. To ensure that the application runs in all browsers, the application needs to be tested in each of the different plug-ins available. Building the models can take place independently of the application development because of the architecture of the application. The Medical Simulation model proved this as it was developed before the actual application was designed. Any model can be integrated into the application by following the guidelines outlined in section 0.

4.3 User-Object interactions

One important attribute of a 3D model is the ability to move objects. For most simulations, this is an important feature and necessary for the accurate simulation of real world events. VRML provides mechanisms for allowing this and for most scenarios it is a simple task to develop the code to move an object. However, because the application is restricted to using 2D controls for input (keyboard and mouse), it is not possible to move objects in 3 dimensions. Clever design can circumvent this problem.

With complex models, there may be many moving parts and sometimes objects must move between other moving parts. For example, in the Office Simulation, the CD in the drawer moves relative to the drawer when it is in the drawer, but moves relative to the CD tray when it is in the tray. This type of movement is more difficult to implement but is nevertheless possible with VRML. Generally, most user-object interactions can be catered for but some are more difficult to code. Complex movements can however be easily coded using animations which are quite easy to build using a VRML editor like Cosmo Worlds.

4.4 Complexity of Models

As the models become larger and more complex, the speed at which the scene is updated slows dramatically. This creates an upper limit on the complexity of usable models that can be developed in VRML. Restricting size, interactivity and complexity of models reduces the authenticity of the model and decreases its learning value. Also, the ability of the user to navigate easily through a VRML model is hindered by the complexity of the world as the refresh rate at which the scene is updated, decreases noticeably as the model becomes more complex.

VRML provides a method of reducing the complexity of models through the use of LOD (Level of Detail) nodes. When these nodes are used it is possible to define representations of an object as it should be rendered depending on its distance from the user. If the user is very far away, a simple shape may suffice. As the user gets closer to an object, more complex representations can be rendered right up to an extremely accurate model when the user is very close to the object. Using level of detail means that rendering can take place much quicker when the models are far away from the user. Only when the user gets very close to the component does the highest level of detail need to be rendered.

While VRML can produce very detailed models, it is sometimes better to use 2D bitmaps as texture maps in combination with 3D models to produce realistic looking models. The benefit of this is that the size of the model remains small, yet detailed. In general, 3D models are of most benefit in VRML, when examination of the 3D structure is a vital component of the training methodology. Areas where it would be of particular use are in biology – where bone structure and anatomical details can be studied and examined, and in chemistry where molecular structures can be easily represented and manipulated by the user.

4.5 Training

The Medical Simulation demonstrated the use of the application in the context of testing the users knowledge in a real task. A typical use for the application and the model would be as part of a Human Anatomy Training course. The 3D environment was most useful for this task as it enabled the user to examine the model in detail and from multiple perspectives. Because of the nature of the task, it didn't have as much emphasis on the constructionist mode of learning. The Office simulation was much more representative of the type of environment in which the user must explore and interact with objects to perform the task. Some knowledge of the technical issues and details of the implementation must be understood when choosing the tasks to develop. This is not always possible in practise, as the person defining the task context or even the task details is not the same person building the simulations.

5 Conclusions and Future Work

The application developed for this project was a success in the sense that it met all the objectives and demonstrated how task-based training can be applied to 3D models. The technologies used in the application allow it to be run over Internet environments and the size of the models are sufficiently small as to allow it to work across low bandwidth connections. One minor drawback is the differences in the implementation of Java and VRML between different browsers. This makes the development of any Java applet more time-consuming and difficult, as it requires the application to be constantly tested in all different configurations.

Because the models are described in text format, it is easy to reuse parts of one model in another. If necessary, objects can be easily scaled or rotated to fit in with other objects. This is a distinct advantage over applications that use 2D images. Generally, 2D images of objects are not displayed at the same scale and the user can loose the sense of relative size of objects. The most suitable area of use for this application is for training where knowledge of a 3D structure of an object is vital. Users can manipulate and interact with 3D models and the application can provide feedback to the user based on their input. Examples include chemical molecules, internal organs and architectural structures.

Navigation though VRML models is a little cumbersome with the interfaces that are programmed into the plug-ins. It is easier to examine objects than it is to navigate around them. 3D games like Quake or Doom have a very user-friendly navigation interface. It should be possible to implement a similar style interface in the application. Some further work can be done with the application to make it more usable and platform independent:

- Allow variable number of steps to be placed in a task
- Viewpoint names are not hard-coded but instead referenced in an INI file or a text file
- Instead of using static HTML pages for the feedback, a better solution would be to have the Java applet dynamically write the HTML at run time
- The Java applet will work correctly on all browsers and with different versions of Java
- The application can provide feedback on each step the user performs. The VRML model keeps track of all the information so it would be quite straight forward to pass the information on to the user
- The Java applet can cater for localised versions of the application

These points are simple enough for a Java programmer to implement. The difficult issues in the design of a 3D task-based training application have already been solved. Overall, the development of the application shows that training in 3D environments is possible using today's technologies, while implementing the latest training guidelines.

Bibliography

[CBT, 1995] CBT Systems, 1995. White Paper on Computer-based Training Methodology, 3

[Cosmo, 1998] 3D Interface Design. Spinning cube example. http://www.cosmosoftware.com/developer/, 12

[Cronin, 1997] Paul Cronin, 1997. Report on the applications of Virtual Reality technology to Education, HCRC, University of Edinburgh, 4

[Dede, 1992] Dede, C.J (1992). The future of Multimedia: Bridging to Virtual Worlds. Educational Technology, 5

[Fox et al., 1994] Geoffrey C. Fox, Wojtek Furmanski, Michael S. Nilan, Ruth V. Small, 1994. Assessing Virtual Reality for Education. Proposal to the National Science Foundation, 6

[Hall et. al, 1998] Craig R. Hall, Randy J. Stiles, Carol D. Horwitz, 1998. Virtual Reality for Training: Evaluating Knowledge Retention. IEEE 1998 Virtual Reality Annual International Symposium, 6

[Isdale, 1998] Jerry Isdale, 1998. What is Virtual Reality? Virtual Reality Information Resources http://www.isx.com/~jisdale/WhatIsVr.html, 4

[Johnson et al., 1998] Andrew Johnson, maria Roussos, Jason Leigh, Christina Vasilakis, Craig Barnes, Thomas Moher, 1998. The NICE Project: Learning Together in a Virtual World. IEEE 1998 Virtual Reality Annual International Symposium, 7

[Kalawsky, 1996] Kalawsky, R.S. (1996) Exploiting Virtual Reality in Education and Training: Technological Issues. Report prepared for the Advisory Group on Computer Graphics (AGOCG), 4

[Lanier, 1989] Jason Lanier, 1989, of VPL Research, 4

[M. Bricken, 1991] M. Bricken, 1991. Virtual worlds: No interface to design. In M. Benedikt (Ed.), Cyberspace: First steps. Cambridge, MA: MIT Press, 5

[Merrill, 1993] M.D. Merrill, 1993. Instructional transaction theory: Knowledge relationships among processes, entities and activities. Educational Technology, 5

[Norman, 1986] D.A. Norman, 1986. Cognitive Engineering. User centered design: New perspectives on human-computer interaction. Hillsdale, NJ; Lawerence Erlbaum Associates, 6

[Ressler, 1996] Anthropometric Landmarks of the Head by Sandy Ressler, Greg Seidman, Matt Buckley. http://ovrt.nist.gov/projects/cardlab/vrmlhead.htm , 11

[Roussos, 1997] Roussos, M. (1997) Learning and Building Together in an Immersive Virtual World, Electronic Visualization Laboratory (EVL) and Interactive Computing Environments Laboratory (ICE) University of Illinois at Chicago, 7

[Spiro et al., 1991] R.J. Spiro, P.L. Feltovich, M.J. Jacobson, R.L. Coulson, 1991. Cognitive flexibility, constructivism and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. Educational Technology, 5

[Warrick, 1996] Philip A. Warrick, VRML: A tool for visualizing anatomy in medical education. Proceedings of The Canadian Medical and Biological Engineering Society Conference (CMBEC 22), Charlottetown P.E.I., Canada, 6

[Youngblut, 1998] Christine Youngblut, 1998. Educational Uses of Virtual Reality Technology, Institute for Defense Analyses, 6

[Zeltzer, 1992] D. Zelter, (1992). Autonomy, interaction and Presence. Presence, 5