

INTELLIGENT GUIDANCE IN A VIRTUAL UNIVERSITY

*T. Panayiotopoulos, , N. Zacharis, S. Vosinakis
Department of Computer Science,
University of Piraeus,
80 Karaoli & Dimitriou str.
18534 Piraeus, Greece
themisp@unipi.gr, spyrosv@erato.cs.unipi.gr*

1. Introduction

Virtual Reality technology [1-3], has introduced a new spatial metaphor with very interesting applications on Intelligent Navigation [4], social behaviour over virtual worlds [5], full body interaction [6], virtual studios [7], etc.

During the past few years, the Virtual Reality Modelling Language (VRML) has emerged as the de facto standard for describing 3-D systems on the World Wide Web. It is platform-independent, easy to use, and gives Web authors the possibility to embed virtual worlds inside their pages. While Java has dramatically altered the way applications are created and distributed, VRML's impact goes beyond, changing the nature of the applications themselves while enriching and deepening the meaning of the data it encapsulates [8]. Recently, a purely theoretical attempt has been made in order to analyse the model-based semantics for Virtual Reality modelling [9]. Another attempt [10] tries to classify the types of behaviour present in Virtual Reality systems.

An interesting application would be the creation of worlds that represent real places or buildings, where the user can be able to access various kinds of information interacting with objects or avatars and travel at the same time in the virtual space. Such applications would require the development of overlying modules which would be capable of communicating with the VRML browsers, providing special control on certain virtual entities, maintaining Information Databases, Interacting with the user, etc. In this system, we have created an interactive application that guides the user inside a virtual university. Visitors can communicate with the program through a

command driven system and have a virtual representation of their requests. More specifically, they are able to “walk” through a virtual building with seven different floors that represents the central building of the University of Piraeus and interact with a guide presented as a human-like avatar. The guide can lead visitors to important places inside the building, according to their information needs, and display the appropriate multimedia documents.

2. Overall Architecture and Interface

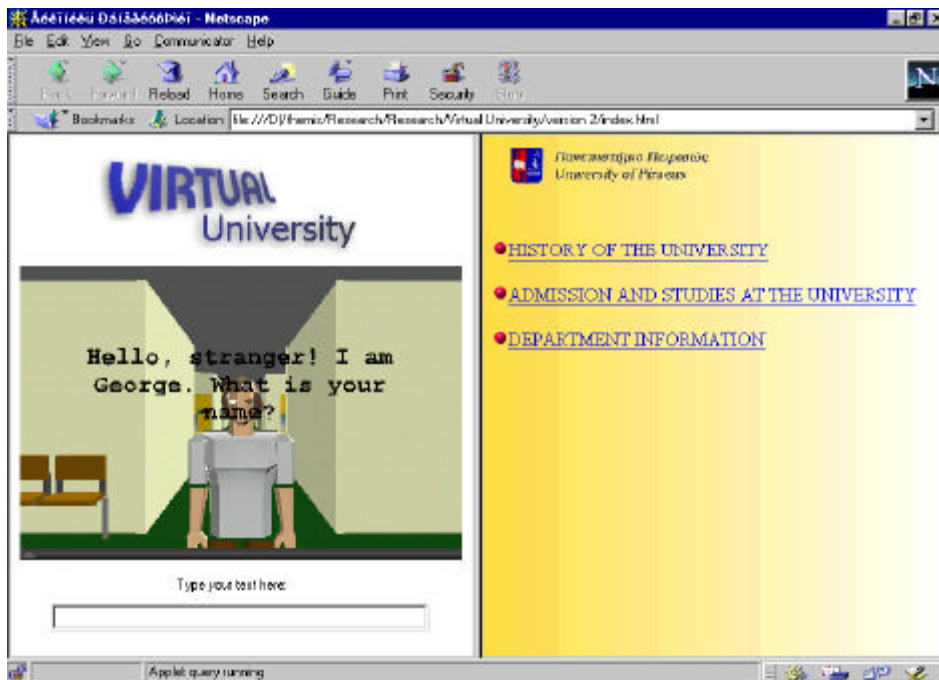


Figure 1. The ‘Welcome’ screenshot of the program

The system consists of several different parts that communicate with each other according to the user’s requests:

- **The User Interface:** It consists of two frames. The first one displays the virtual world that shows the 3-D content as well as a text field where the user can type his/her commands, and the second one is the Information Panel, where the multimedia HTML pages are loaded.
- **The VRML models:** They are subdivided into the static models that represent the main building of the University of Piraeus and the avatar model, which is a virtual representation of a guide.
- **The Multimedia Library:** HTML pages that are displayed in the Information Panel and contain text, image and sound.

- **The Information Database:** It contains information about persons, places, research groups, departments and other entities of the university and also provides links to the multimedia pages.
- **The Spatial Graph:** A graph containing spatial information about the floors of the main building to help the avatar's navigation.
- **The Avatar Control:** The part that is responsible for the avatar's actions inside the virtual world. It uses the spatial graph to implement the movement commands.
- **The Information Control:** The process that searches the Information Database, displays result messages, and loads multimedia pages on the Information Panel.
- **The Command Interpreter:** The part that processes the user's commands, creates a set of actions, and calls the Avatar Control and/or the Information Control to implement them.

When the application starts, it loads a static model representing a floor of the university's main building and the avatar model. It also displays the starting page on the Information panel (Figure 1) and presents a message to the user. According to the user's command it displays multimedia content by calling the information control, or causes an avatar action processed by the avatar control. Furthermore, the user can access information directly by clicking on objects of the virtual world.

The whole architecture of the system can be displayed in the following diagram:

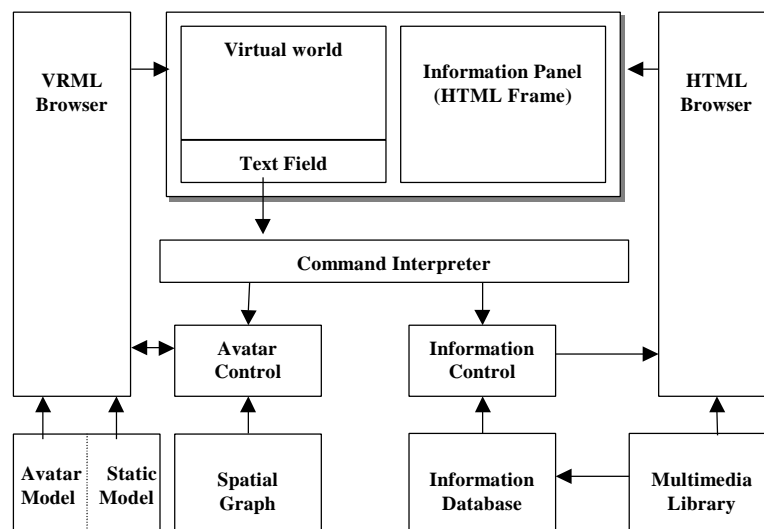


Figure 2. The overall architecture

3. Command Interpreter

When the user enters a command in the text field, the command interpreter tries to recognize it and creates a sequence of actions that are processed by the respective parts. It repeats calling the Avatar Control or the Information Control using specific parameters until the sequence is over, and then it is ready to read the next user command. The commands recognized by the system are divided into four categories:

Movement Commands: They cause a movement of the avatar, represented as a “virtual walk”, that is followed by the user’s viewpoint. These commands are: Goto <floor number>, Goto <office number>, Goto <individual>, Goto <room name>. Their meaning is obvious.

Informative Commands: These commands display information to the user according to his request. The result can be a hypermedia HTML page or simple text.

TellMeAbout <individual name>: Displays information about the individual.

TellMeAbout <office number>: Displays the professors that use the office.

TellMeAbout <course name>: Displays the professors that are teaching that course.

TellMeAbout <subject>: Displays the professors that are related to the subject.

TellMeAbout <department>: Displays general information about a department of the University.

TellMeAbout <room name>: Gives information about other interesting rooms, such as the library, laboratories, etc.

Compound Commands: These commands use a combination of movement and information giving to display more complex actions.

Tour <department>: The avatar guides the user through a whole department, providing information about the important places.

Tour <floor>: Guidance in the specified floor

Tour Library: The avatar presents the Main Library of the University

Various Commands: They perform various actions, such as:

ReturnToGuide: The user’s viewpoint returns to the position of the guide

Help: Shows on-line help

About: Information about the system

Stop: Terminates instantly and returns control to the user.

4. The Virtual Guide

4.1 THE VIRTUAL UNIVERSITY

The university model has been created using the ground plans of the real building. The walls and the additional objects were placed in the virtual world by translating their real coordinates into the respective 3-D coordinates of the model. There is a basic VRML file that contains the avatar model, the lights and the viewpoints and seven

different worlds representing the floors that are called from the basic world using a Switch node. We have used this architecture, instead of having a single world for the whole building, in order to lower the complexity of the VRML models and achieve better performance. Otherwise, the program would require huge memory and increased processor speed and it would be impossible to run in average home computers. Whenever a different child Node is selected in the Switch node, the browser loads and displays the new floor without affecting the main world.

4.2 AVATAR MODEL

The avatar is a virtual human that is presented to the user to help him navigate inside the building. It can display phrases, walk from any place to another, change floors, and load various information on the screen. It consists of two separated parts: the **avatar model** that is responsible for its visual representation and the **avatar control** that is responsible for its behaviour.

The avatar model is a VRML object that presents a human being plus a set of Interpolators used to simulate the movement of its hands and legs during the walking process. The model is divided into seven different parts according to the avatar's limbs. There are six Interpolators to change their orientation and one Time Sensor to control the animation. More specifically, the avatar's body parts are:

- the main body and head
- the right arm
- the left arm
- the right leg
- the left leg

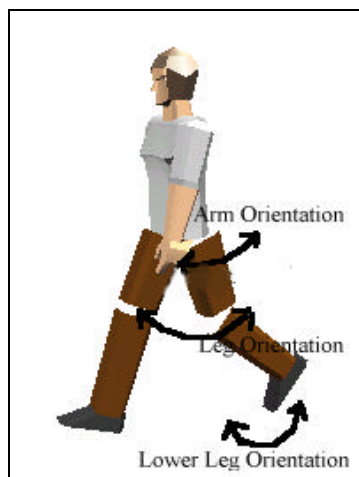


Figure 3. Orientation changes of avatar's body

Each leg is subdivided into its thigh and shin. The position of the shin is relative to that of the whole leg, so changes in its orientation do not affect its position in the body. Furthermore, all the limbs of the avatar maintain their relative positions, therefore a movement of the avatar does not require independent movement of all body parts.

The avatar's limbs have been routed to six independent OrientationInterpolators. An OrientationInterpolator interpolates between two orientations by computing the shortest path on the unit sphere between the two orientations. The interpolation is linear in arc length along this path [2]. The orientation changes in the avatar's body have been depicted in figure 3. There are three interpolators for each side – left and right - controlling the movement of the arm, the leg and the shin.

Finally, all the interpolators are triggered by one TimeSensor, which loops every two seconds. Each time the sensor is enabled, the avatar starts animating by moving its hands and legs. This animation starts whenever the avatar changes its translation to simulate walking.

4.3 THE WALKING PROCESS

The next task that the avatar can perform is “walking”. The avatar control uses a set of functions to control the avatar's movement and rotation, and to make sure that it follows the shortest path without colliding to solid objects. The most primitive function is the one that transports it from one point to another using linear movement. First of all, taking into account the current coordinates of the avatar and the coordinates of the “target”, it calculates the angle φ that is defined by the horizontal line and the line that passes through the two points, as shown in the following diagram:

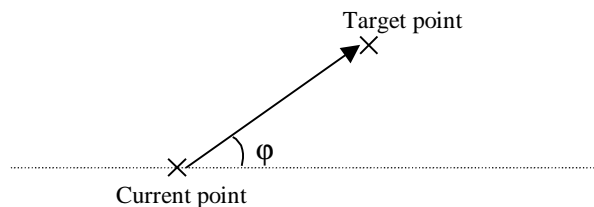


Figure 4. Rotation of the avatar

After that, the program changes the rotation of the avatar, so that its orientation in the two dimensional horizontal field is φ . A new thread is started, that makes the avatar move with constant speed by constantly changing its position using $\sin(\varphi)$ and $\cos(\varphi)$ to preserve its route. For example, if the desired speed is 2 meters per second and the thread changes the avatar's translation every quarter second, then the avatar must be moved by half meter after each time fraction. The new values for x and z coordinates of the avatar are as follows:

$$x = x + 0.5 * \sin(\varphi)$$

$$z = z + 0.5 * \cos(\varphi)$$

Each time the avatar makes a move, the new coordinates are compared to the target coordinates, and when the distance between them is less than the distance covered by a time fraction, the system assumes that the destination is reached and places the avatar in the correct position.

When the thread starts for the first time, the program enables the Time Sensor that creates the animation of “walking” and the avatar’s hands and legs are constantly moving during its transportation.

This function is not enough to simulate the avatar’s movement. It must also be able to avoid any solid objects that stand in its way. Therefore, the avatar control uses a spatial graph of the floors of the University to create a “safe” path for the avatar to follow. Each floor has been assigned to a complex undirected graph, the nodes of which store the three-dimensional coordinates of the corresponding position in the Virtual world as well as some additional information. The “avatar control part” ensures that the avatar walks only between the nodes of the spatial graph, so that a possible collision between the virtual guide and an object of the static world is avoided. Nodes are divided into two groups:

Intermediate nodes that exist only to provide a path for the avatar’s movement.

Termination nodes, that correspond to virtual places with access to informative links. Only these nodes can be requested from the user as a target, and once they are approached, they provide an appropriate link to the multimedia library as well as the information database, so that a relative content is displayed in the information panel.

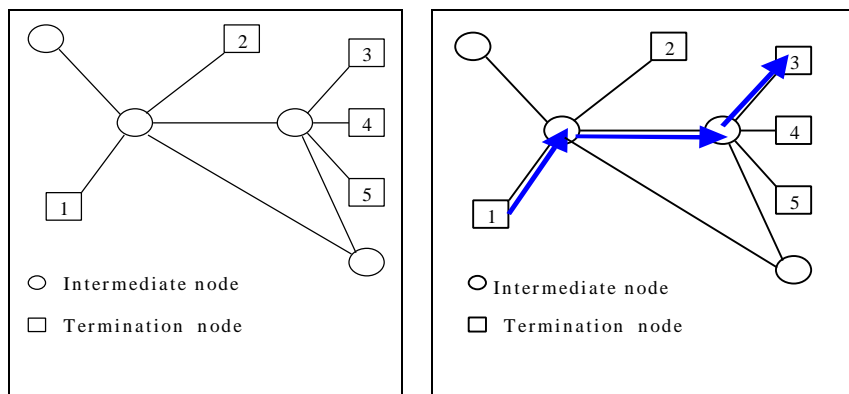


Figure 5: (a) An example of a spatial graph (b) A route between two nodes

Whenever the avatar has to reach a certain destination, the spatial graph is loaded, and the program tries to find the shortest path between the current position and the desired

node. The route is planned, and the avatar starts walking through the selected path. For example, if the user requested a walk to the node 3, and he was currently located at node 1, the avatar's route would be as described in figure 5b.

4.4 CAMERA CONTROL

The avatar control is responsible for the movement of the guide in the virtual space, but it has to do more than just changing the translation of the avatar. The user's viewpoint must also be moved to have a constant visual representation of the walking process. Therefore, we have used three different cameras to control the animation and to ensure that the user's view of the scene is correct. The system uses:

Two cameras attached to the avatar: they both have the height of the average human eye – at about 1.6 meters. The first one (CAMERA A) is located two meters behind the avatar and the second one (CAMERA C) two meters in front of the avatar.

One independent camera: It also has the height of the human eye, and can be placed anywhere in the scene (CAMERA B).

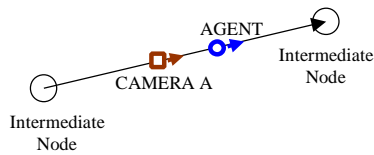


Figure 6. Diagrammatic representation and output of Camera A

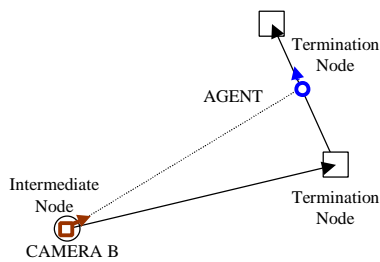


Figure 7. Diagrammatic representation and output of Camera B

When the animation starts, the user views the output of the first camera, which automatically follows the avatar's movement. Whenever the avatar moves from an intermediate node to a termination node or between termination nodes, the independent camera is placed at the current position of the user, maintains its translation, and changes constantly its orientation so that the avatar is always at the center of the screen. After that, when the avatar has reached its destination, it rotates 180° and the third viewpoint (CAMERA C) is enabled, so that the user looks straight at the avatar and is able to read its messages.

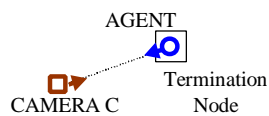


Figure 8. Diagrammatic representation and output of Camera C

5. Implementation

The Virtual University system has been implemented using interactive VRML 2.0 worlds, a Java applet and HTML pages. It runs inside a single page of a simple Web browser with Java capabilities and a VRML 2.0 plug-in. The page consists of two basic frames, one for the VRML world and Java applet, responsible for the 3D content and the user interaction, and one for the HTML pages that display the requested information.

As far as communication between a VRML world and its external environment is concerned, an interface between these two is needed. This interface is called External Authoring Interface (EAI) and it defines the set of functions on the VRML browser that the external environment can perform to affect the VRML world [11]. The EAI allows a currently running Java applet to control a VRML world, just like it would control any other media [12].

6. Conclusions and Future Work

In this chapter we have presented the architecture of an intelligent guidance system inside a virtual environment. It is an effective method for easily accessing the desired information from a huge amount of pages using virtual reality techniques and travelling at the same time in a representation of a real building. It is based on the latest features of the World Wide Web, and therefore it is multi-platform and can be accessed by a number of users while running on a single server.

We are currently working on extending the system's interactive capabilities and making it more attractive to the common users. Moreover, we are planning to add multi-user support using avatars to represent the users, and to put more intelligent agents [13] as well as other interactive objects such as elevators, doors, computers, etc.

Acknowledgement

The system described in this chapter was partially funded by the EPEAEK project (EKT., Subprogram 3., Measure 3.1., Action 3.1.B) entitled "Modernisation of the Central Library of the University of Piraeus", funded by the European Community and the Greek Ministry of Education and Religious Affairs.

References

- [1]. J. Vince, "Virtual Reality Systems", ACM Press, 1995.
- [2]. The VRML Consortium Incorporated., "VRML97 International Standard", (ISO/IEC 14772-1:1997) <http://www.vrml.org/Specifications/VRML97>, 1997
- [3]. S.Vosinakis, T.Panayiotopoulos, 'State of the Art in Virtual Reality', Internal Report, University of Piraeus, Dpt. Of Computer Science, 1997 (in Greek).
- [4]. N. Zacharis, T. Panayiotopoulos, "A Learning Recommendation Agent in – Virtual Environments", International Conference for Artificial Intelligence and Soft Computing, Cancun, Mexico, 1998.
- [5]. Y Honda et al, "Virtual Society: Extending the WWW to support a multi-user interactive shared 3D environment", Procs VRML 95, San Diego, 1995.
- [6]. P. Maes, et al, "The ALIVE system: full-body interaction with autonomous agents". Proceedings of Computer Animation '95, 1995.
- [7]. S. Gibbs, C. Arapis, C. Breiteneder, V. Lalioti, S. Mostafawy, J. Speider, "Virtual Studios: An Overview", IEEE Multimedia, pp.18-35, Jan – Mar 1998.
- [8]. C. Marrin, B. McCloskey, K. Sandvik, D. Chin, "Creating Interactive Java Applications with 3D and VRML", <http://cosmo.sgi.com/developer.html>, Silicon Graphics, 1997.
- [9]. M. Prokopenko, V. Jauregui, "Reasoning about actions in Virtual Reality." IJCAI-97 Workshop on Nonmonotonic Reasoning Action and Change, 1997.
- [10]. B. Roehl, "Some Thoughts on Behavior in VR Systems (Second draft: August, 1995)", URL: <http://sunee.uwaterloo.ca/~broehl/behav.html>, 1995
- [11]. C. Marrin, "Proposal for a VRML 2.0 Informative Annex. External Authoring Interface Reference", <http://cosmo.sgi.com/developer.html>, Silicon Graphics, 1997.
- [12]. J. Doppke, D. Heimbigner, and A. Wolf, "Software Process Modeling and Execution within Virtual Environments", ACM Transactions on Software Engineering and Methodology, Vol.7, No.1, pp. 1-40, January 1998.
- [13]. T. Panayiotopoulos, G. Katsirelos, S. Vosinakis, S. Kousidou, "An Intelligent Agent Framework in VRML worlds", Third European Robotics, Intelligent Systems & Control Conference, EURISCON'98, Athens, June 1998.