Using Virtual Reality Techniques for the Simulation of Physics Experiments

Nikos Avradinis, Spyros Vosinakis, Themis Panayiotopoulos
Dept. of Informatics, University of Piraeus,
Knowledge Engineering Laboratory,
80 Karaoli & Dimitriou Str, 18534 Piraeus, Greece.
avrad@unipi.gr, spyrosv@unipi.gr, themisp@unipi.gr

ABSTRACT

Educational software has known significant evolution over the past few years due to the introduction of technologies such as multimedia and the World Wide Web. These innovations have significantly offered in the field of computer-based instruction, by providing the means for remote, interactive and intuitive applications. There are, however, some categories of applications such as Virtual Laboratories, where these applications prove to be inadequate. In this paper we discuss some of the limitations of multimedia and the world wide web and present some of the advantages virtual reality technology has to offer in the field of education. We also propose an architecture and methodology for the development of Virtual Laboratory applications and present a case study on the development of a Virtual Physics Laboratory.

Keywords: virtual reality, interactive learning, educational systems, virtual laboratory, intelligent systems

1. INTRODUCTION

During the past few years we have witnessed significant evolution in the computer industry mainly due to the introduction of multimedia and the World Wide Web.

The innovative features of multimedia and the World Wide Web quickly found application in the field of education [1], [2], [3]. The use of Multimedia and Internet technology led to the presentation of a plethora of various widely recognized educational systems. It is quite true that modern technology has provided the means for accessing very quickly a vast amount of information and is capable of presenting it through impressive user interfaces. However, there seem to be a whole category of applications with high visualization needs where multimedia technology was not as effective as expected.

Recently, there have been attempts to integrate Virtual Reality (VR) with educational software to produce a higher level of interaction and visualization. Researchers have started attempts to model concepts such as 'Virtual Classroom' [4], three-dimensional representations of chemical structures [5], 'Virtual Laboratory' [6], [7], 'Virtual University' [8], etc. It seems that VR provides a new approach to learning as it increases the interest and provides in this way an alternative educational process.

Nevertheless, there are issues to be considered for the development of effective virtual reality applications. A virtual environment has its limitations and should be under some control. Artificial Intelligence, on the other hand, can offer significant help by providing the missing link of behavioral control. It has been proved that Logic Programming architectures can be effectively connected to virtual environments [9], enhancing their semantics.

In this paper we present the architecture of a virtual laboratory, a three-dimensional world, where the students are able to interact with the learning environment and perform their experiments in real time. The user is embedded in the virtual world and can change the position and orientation of its objects, carry objects around, or put objects together and let them interact, e.g. by connecting a device to the power source. Virtual objects respond to the laws of physics in a natural way and generate the appropriate results according to the current state of the system.

The proposed architecture of such a system consists of three different parts. At a low level lies a logic component, which is responsible for the physical-based modeling, that is, the application of the laws and principles that refer to the experiment's world. The second module is the 3D virtual engine, which handles the visual representation of the laboratory and creates the user's view according to his/her position in the virtual space. The third module is the interface, which takes care of the human-computer interaction within the laboratory, interprets the user's actions and creates the data for the logical component. Each module of the system is implemented by different technologies (Prolog, HTML, Java and VRML) integrated into a single system producing a controlled, interactive experimentation environment. We also present an implemented example of a physics laboratory, where the user can experiment with the laws of gravity and kinematics.

2. CLASSIC MULTIMEDIA TECHNIQUES - ADVANTAGES AND LIMITATIONS

Multimedia technology allowed the creation of a new generation of software development tools, with sophisticated media integration and handling capabilities. This enabled software designers to create attractive and user friendly interfaces, that encourage use of the product and greatly increase user involvement. Multimedia technology facilitates the adoption of approaches such as multisensory education, allows users to interact with the system in various ways and receive not only textual, but also video or audio feedback.

Video, audio, animation and images also make possible the production of highly memorable, illustrative explanations of concepts [10], and break the barriers of verbal communication. The use of artificial intelligence techniques for the production of intelligent multimedia software further enhances interaction and makes applications more intriguing and appealing to the user. All of the above lead to better knowledge retention on the part of the user.

The World Wide Web also presented educators with a new educational medium of immense potential, yet unknown to its full extent. Remote access to educational resources was made possible, which eliminated space limitations, as students and teachers no longer had to restrict themselves to material belonging to their own school or library. Time restrictions were also eliminated, as web servers do not follow office hours; educational content can be delivered twenty-four hours a day, seven days a week. The use of widely accepted standards offered platform independence, relieving software developers from the need to adapt applications to various computer platforms. The notion of hypermedia, with billions of links between related documents spanned over the whole world allowed the creation of networks of related concepts, essentially turning the web into a vast, universal knowledge base.

We will not argue any further on the potential of Multimedia and Web technology for the development of computer based instructional applications. Their value has already been proven by a vast number of applications and research studies. We are going to focus on some of the limitations of classic multimedia[1] techniques when it comes to specific categories of computer applications.

Classic multimedia seems to be very well suited to applications serving mainly reference material, or applications that require more or less simple visual representations. There seems, however, to be a whole category of applications where the classic multimedia approach provided by the above methods is proven inadequate. Geometry, geography, history, chemistry, biology and physics are all domains which greatly depend on visual information for user instruction. Dealing with the natural world and trying to model real-world phenomena, these domains call for three-dimensional representations, rather than the classic two-dimensional approach. Similar situations cannot be dealt with the use of classic multimedia or Web techniques, as they provide tools only for 2D objects (images and video). There are some multimedia applications that use three-dimensional presentations, in the form of animations. These, however, are non-interactive and only allow users to watch a static presentation, rather than let them take control and guide themselves through the system.

Another serious limitation is the fact that traditional multimedia applications adopt a third-person point of view approach. Users are not considered to be an active part of the system. They are instead treated as external (elements) agents, who communicate with the system utilizing indirect methods, verbal or iconic in most cases. This can easily deter users from getting involved with the application, thus leading to failure to achieve educational objectives. The third-person point of view approach is to an extent responsible for the fact that interaction is still limited. Although greatly improved in comparison to textual applications, inter action in traditional multimedia applications is restricted to either mouse functions or commands that in most cases serve navigational needs.

This also relates to another observation, very common among educational software. The vast majority of desktop multimedia applications can be treated as media-rich electronic books, providing mainly reference material, rather than creating a virtual educational environment that will allow the student to interact and experiment with.

Most of these restrictions also apply to HTML-based WWW applications. The level of interaction in Web-based applications is rather low, due to limitations of the medium itself. The user communicates with the system either by mouse clicks to activate hyperlinks or by simple text entry. Although efforts have been made to increase interactivity on the Web with the use of scripting languages or mobile code, these have many technological limitations and are restricted to 2D-only interaction.

[1] We will use the term “classic” or “traditional” multimedia to denote 2D multimedia applications, in contrast to 3D virtual reality applications.

3. VIRTUAL REALITY AND EDUCATION

Many of the limitations of classic multimedia can be surpassed with the use of Virtual Reality technology. Virtual reality incorporates characteristics that lend it significant potential: immersion, presence, direct engagement (user involvement), immediate visual feedback, autonomy and interactivity [11], [12], [13]. These characteristics, along with the fact that VR inherently supports three-dimensional modeling, make it almost ideal for specific types of applications [14].

In virtual reality worlds, the user is no longer treated as an external viewer. He/she is actually a part of the system, an autonomous presence in the virtual world. He/she is free to navigate around the virtual environment, move in three dimensions, interact with objects, look behind or under them and examine the world from different viewpoints, which is not possible in classic two-dimensional multimedia. Media integration is even higher than in desktop multimedia, with the use of three-dimensional audio and haptic feedback.

Moreover, virtual reality allows the simulation of dangerous or expensive environments. It also allows the creation of synthetic worlds, which gives a whole new perspective for the development of various kinds of applications [15].

These features of virtual reality technology fit very well to educational applications. The three-dimensional representation model is an important feature. Objects in a virtual world are represented much more accurately than 2D-objects, as the missing dimension of depth is added. More accurate illustration is also achieved through the option to observe the virtual world from various angles of perspective, which is impossible in 2D worlds.

Immersion helps retention, by making important concepts more memorable, while also encouraging use of the learning environment [16], [17]. The higher level of interaction also encourages use of the environment, as users are actually obliged to participate in order to receive feedback.

One has to make an important notice, however, regarding the issue of interaction. Interaction is imposed to the user, as it is required for even the simplest task in a virtual world. However, true interaction comes together with feedback from the world. Without meaningful object response to user actions, interaction is restricted to navigational tasks only. There is a need for a high level of autonomy, by the creation of objects with embedded behavior.

This shows the need for virtual reality technology that does not simply allow the creation of impressive 3D interfaces. It should in addition provide the means to embed some kind of behavior to virtual objects, for example by allowing virtual world designers to "program" objects to respond in some way when they receive stimulus from the user or the world itself [18].

4. THE VIRTUAL LABORATORY CONCEPT

A challenging educational application, is the creation of Virtual Laboratories, synthetic worlds where the user can conduct experiments and observe the results.

Virtual Reality technology suits very well the needs of sciences that require a higher level of visualization and interaction. For example, an application for learning and experimenting with Stereochemistry needs 3D technology to display the molecules of a chemical structure. Especially in the case of Physics, Virtual Reality allows the simulation of experiments conducted in a 3D-world, in contrary to the common 2D-only approaches. A three dimensional environment is also necessary in cases like thermodynamics or kinematics, where the visualisation process requires an extra dimension.

In addition, experiments dangerous or too expensive to conduct in an actual laboratory can be simulated with the use of Virtual Reality techniques, like in the case of aerodynamics or nuclear physics. One can also simulate experiments impossible to be conducted in actual laboratories, using laws that either do not apply in the real world, or apply in distant environments only (eg. Gravity on the Moon).

The Virtual Laboratory concept has already been proposed by various researchers [19], [20]. Each one, however, defines the term "Virtual Laboratory" in a different way. Reed & Afjeh adopt a 2D-approach, using HTML and Java to create an environment where interaction is based on standard HTML/Java elements. Firmeza & Ramos propose the use of a hybrid 2D/3D approach, where a 3D VRML model is used to display the laboratory itself, while users perform the experiment and interact with the system using CGI script or Java-enhanced HTML pages. Dede et al. follow a more sophisticated solution, using a high-end Silicon Graphics Onyx RE/2 server combined with 3D visualisation and haptic navigation and feedback hardware. The synthetic world they have created is fully three-dimensional. The user is immersed in the 3D world and performs all actions interacting both with menus and the 3D objects themselves.

We propose that a Virtual Laboratory should present the user with a synthetic world where he/she will have the greatest possible extent of freedom, so that he/she can freely experiment in real time. If any control or restrictions are set by the system, these should affect only navigational tasks, in order to prevent disorientation and simplify movement around the synthetic world (e.g. collision detection). The synthetic world should consist of a virtual space with all the necessary tools to conduct an experiment. The user must be able to combine these tools and learn from his experimentation. Whether such an application will adequately serve educational needs depends on the realistic behaviour of the objects, the precision of the visual data, and the level of complexity that has been used to represent the natural laws or principles governing the specific field.

An Architecture for the development of Virtual Laboratories

We propose an approach for the development of a virtual laboratory. The architecture of our system consists of three different modules:

- The core of the application is the **Logic Component and Physical Based Modelling** part, which is responsible for the behaviour of the virtual objects and the conduction of the experiments. It controls the objects' attributes (e.g. position, velocity), checks for interaction between them and applies the natural laws. After receiving the initial data, it repeats calculating the intermediate values of the objects until the experiment is over.

  Using mathematical models of the natural laws the system is moving from the current state to the next one after a minimum time fragment, which is supposed to be almost equal to zero. The selection of the time fragment Ät depends on the scientific field of interest and the desired accuracy of the conducted experiments. One other important factor that should be considered in this selection is the time that the system needs to update the visual data. If this time is greater than the selected Ät, the experiments will not be visualised in real time. During the transition from one state to another, the Physical Based Modelling part must check all conditions that could cause the application of a natural law, and apply the results to the affected objects, e.g change their attributes respectively.

- The visual representation is created by the **3D Virtual Engine**. It displays all the three dimensional data that is required for the user's view, according to his/her position in the virtual space, and has also the ability to create or destroy 3D models dynamically and transform them in real time. While an experiment is running, it receives values directly from the Logic Component and Physical Based Modelling part, and updates the visual attributes (position, orientation, scale, material) of the objects that take part. The 3D engine must have the ability to interpret widely accepted file formats in order to allow the creation of the objects from the powerful 3D modelling packages.

- The last part that we use in our proposed approach is the **Interface**, which controls the user's navigation and interprets his/her actions. It is responsible for the human-computer interaction, and the generation of the appropriate data that will be transferred to the 3D Virtual Engine and to the Logic Component and Physical Based Modelling part. Whenever the user is moving or interacting with an object, the Interface must be able to handle this action and visualise the results, either using the 3D Virtual Engine or its own text messages.

Once the user enters the system, his/her navigation and action commands are sent to the interface. When an action is performed, the interface feeds new data to the 3D Virtual Engine, and the output is directly displayed to the user. The initialisation of an experiment causes the Interface to communicate with the Logic Component and Physical Based Modelling and send the initial values of the objects. Then, while the experiment is running, changes in the state of the objects are displayed through the 3D Engine, and the user is still free to navigate. The architecture of this system is depicted in the following diagram.

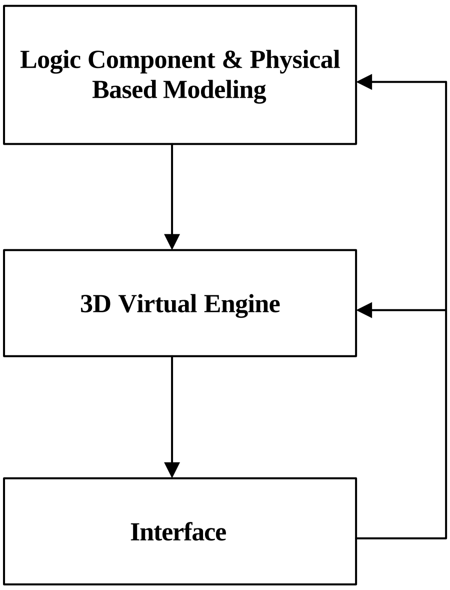

Figure 1 : The architecture of our system

Design and Development-Issues to be considered

Several issues have to be taken under consideration during the design process of a Virtual Laboratory application. The whole design of the virtual laboratory heavily depends on the field of interest, as it will determine the objects needed to be modelled, the range of experiments that will be simulated as well as the laws and principles that will govern the experiment. Initial and terminal conditions for every experiment have to be defined. Subsystems of the virtual laboratory architecture will have to be implemented according to these considerations. The interface should also be designed according to the specific needs of the experiments that will be conducted, as well as the target users of the application. The system will finally have to be tested on a number of users in order to receive feedback about its efficiency and possible modifications that have to be made.

The proposed methodology for the creation of virtual laboratory applications follows:

1. Initially one has to define the field of interest, that is the scientific field that will be used for experimentation (Physics, Chemistry, etc.)
2. Then the basic elements that will take part in the experiments have to be defined. The selection process must be based on the scientific field that is chosen in the previous step. These elements may be modelled using an object-oriented approach, where their attributes and methods have to be declared. In addition, their visual representations could be created using a 3D-modelling package.
3. The next step is to define the natural laws that will be used in the experience. The conditions for the application of each law and its effects on object's attributes have to be declared.
4. The termination condition, e.g. the condition that the system will check to decide whether the exp eriment has finished or not, should be defined.
5. After that, the Logic Component and Physical Based Modelling part has to be developed and tested with initial values to assure the correctness of the results.
6. The last part is the design and development of th e Interface according to the knowledge and experience of the target users. One has to define the set of actions that the user will be allowed to perform and the degree of freedom that he/she may have inside the Virtual Laboratory.
7. The application has to be tested on a large number of users receiving feedback about the difficulties that they encountered, and the respective parts have to be modified.

5. IMPLEMENTATION

Various approaches and technologies for the implementation of a virtual laboratory application can be adopted. One possible approach to implement the system described above is using Prolog for the logical core, Java for the interface and physical based modeling, VRML for the 3D scripting and visualization, and HTML for all other multimedia educational pages (2D visualization) and the integration of the system.

The Virtual Reality Modeling Language (VRML) is a powerful tool for the description of virtual worlds in the World Wide Web. VRML is capable of representing static and animated dynamic 3D and multimedia objects with hyperlinks to other media such as text, sound, video, and image, while VRML browsers, as well as authoring tools for the creation of VRML files, are widely available for many different platforms. On the other hand, 3D models and animations created by well-known software such as Kinetix 3D Studio or NewTek Lightwave 3D can be exported to VRML files and become available to the Web users, providing that they have a VRML plug-in installed in their Web browser. In VRML one can set a number of predefined animations and use various sensors to trigger them, providing a simple way of user-interaction. A lot of interesting VRML worlds concerning education, entertainment, science etc. can be found in the World Wide Web, but most of them are static and have little interaction.

The key to successful virtual worlds is the programmable behavior of the 3D objects provided by VRML's ability to be controlled by external programming languages. One of the most popular approaches is the External Authoring Interface (EAI), a set of Java packages that allow an applet [21] to control one or more embedded VRML worlds in the same HTML page. It gives an applet the ability to read or change the attributes of the virtual objects, as well as to create or destroy VRML nodes dynamically, and all that in real time, while the user is navigating in the virtual space, or using the applet's components. In the last couple of years the External Authoring Interface has been widely accepted by the developers, and most VRML 2.0 plug-ins have EAI support.

On the other hand, Logic Programming files can be efficiently used in Java applications. An example of such an interconnection is the combination of SICStus Prolog and Jasper, a package containing a set of Java classes which can be used to create and manipulate terms, ask queries and request one or more solutions. With the use of a Prolog runtime environment that supports Java, a logic programming module can be inserted in a Java/VRML application and enhance it with features, such as natural language processing, intelligent agents etc. that can be easier implemented using logic rather than procedural programming.

6. AN EXAMPLE APPLICATION: A PHYSICS LABORATORY

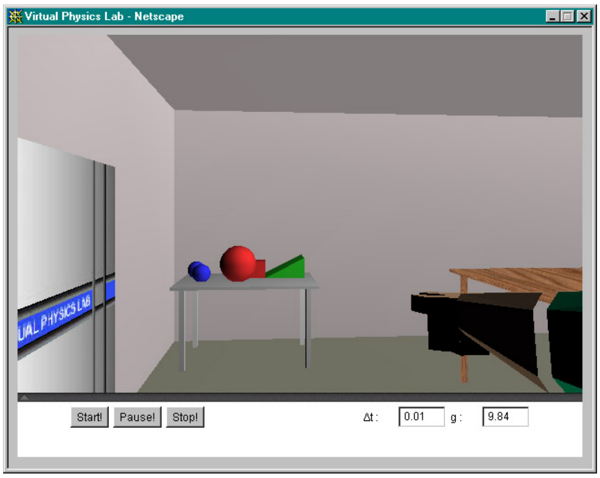

Figure 2 : A screenshot from the Virtual Physics Laboratory

Following the design and development methodology previously presented, we developed a physics laboratory example. The user can navigate in the three-dimensional space, combine the available objects and experiment with the laws of gra vity, friction and kinematics. To be more specific, the application displays a virtual room with a set of primitive objects (spheres, boxes, surfaces, etc) and an experimentation table, where the user can place the selected objects and view the experiment. After the user has selected the objects to interact with each other, and has defined initial conditions, the experiment starts with the press of a button, and the objects are interacting with each other in real time. When the experiment is over, either because it has finished or because the user wished so, the system is ready for a new experiment to be set-up.

The actions that the user can perform inside the Virtual Physics Laboratory are:

- *Navigation*: The user can walk or rotate inside the Virtual Environment using the VRML browser's navigation toolbar.
- *Object carrying*: A simple mouse click on an object moves it to the "virtual hand" of the user, and he/she can carry it around the laboratory while navigating. A second click leaves the object floating in its current position. Objects do not respond to the laws of gravity before the experiment actually starts with a user's command.
- *Setting initial object attributes*: The user can set the initial velocity or acceleration of an object by selecting it.
- *Setting global attributes* : The minimum time fragment Ät and the gravity acceleration g can be set using the interface.
- *Starting, stopping or pausing an experiment*: Using the interface buttons the user can choose to start, stop, or pause an experiment.

The case study has been implemented using a set of different technologies. We have used Java to develop the physical based modelling part and the interface and Virtual Reality Modelling Language (VRML) for the visualisation of the experiment. The whole project is integrated in a single HTML page and can be viewed using a Java-enhanced web browser (such as Internet Explorer or Netscape Communicator) with a VRML 2.0 plug-in. The communication between the Java applet and the VRML world is implemented with the use of the External Authoring Interface (EAI).

The primitive objects have been modelled in Java classes (*pObject* for spheres and boxes, and *pSurface* for surfaces), and use VRML 2.0 files for the 3D representation. Each class contains attributes about the object's physical attributes, such as velocity, mass, size, material, acceleration, etc, and once it is initialised, the correct values are set. Each physical object has its own material, which is an instance of a new class (*pMaterial*) that contains information about the material's physical (e.g. density) and visual (e.g. texture image) attributes. Vector values (e.g. velocity) are using an instance of the vector class (*pVector*). There is also one class responsible for the whole experiment (*pExperiment*), which contains the set of objects that is being used, and all the methods that handle the details of the physical based modeling. Below is a Class diagram of the basic classes used in our system.

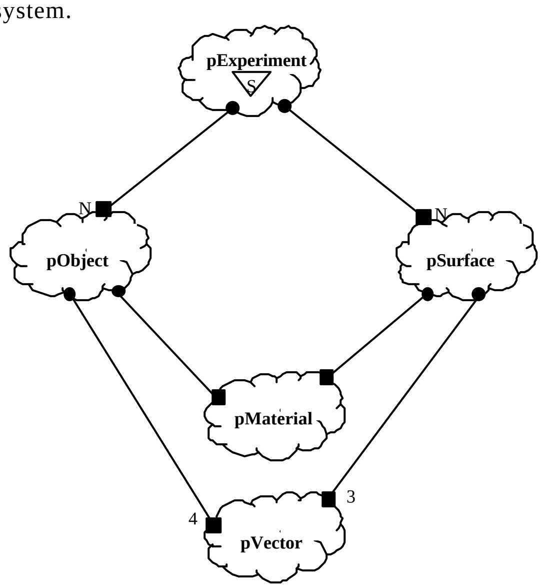

Figure 3 : A Class Diagram of the Physical Based Modeling module

When the user decides to start an experiment, a new thread is initialised, which, starting from the initial values of the used objects, calculates the next values according to the given time step Ät, and checks for any interaction between them that might have an effect on their attributes. The system decides whether two objects have collided with each other, or whether one is lying on another, according to their distance, their velocities, etc. When a certain type of interaction is diagnosed (friction, crash, gravity, etc), the appropriate rule is applied, and the new values are set. When this process is over, the system checks whether the experiment is over (velocitiy and coefficient force on every object equal to zero) or not. In the second case the system advances to the next step, adding Ät to the current time. Meanwhile, it communicates with the VRML browser using the External Authoring Interface (EAI), and updates the world as necessary. In our example, it updates only the position of the objects, but in another case (e.g. a chemistry lab) changes could also affect their colour, size or geometry.

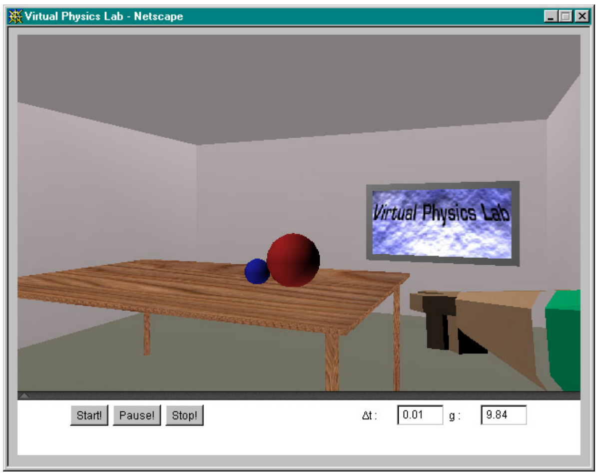

Figure 4 : A collision between two balls

7. CONCLUSIONS

In this paper we have seen that Virtual Reality in education can be a valuable tool for special kinds of applications. We presented the methodology, architecture and implementation of a virtual laboratory, and demonstrated our own example, a Physics Laboratory. This application can of course be modified for other fields of interest.

The performance of the 3D engine that we used (VRML) is client dependent, and therefore does not impose any burden on the part of the server. There are, however, some requirements for the client system, as the Java / VRML combination produces results that depend much on the amount and complexity of the 3D models used, and a complex virtual world would require high-end equipment to have an acceptable performance. So, one difficulty that we encountered was to maintain a balance between quality and performance, providing that our application is Web based and platform-independent.

Another important point that should be mentioned is the difficulty of navigating in three dimensional environments using 2D devices (like keyboard, mouse or joystick), a fact that can cause a lot of confusion to the non-experienced user. Nevertheless, we believe that today's expensive 3D input and output devices (e.g. head mounted displays, haptic devices, etc.) will be tomorrow's standard features of an average home computer.

We are currently extending our case study by introducing new objects and supporting a greater number of physical laws to experiment with. Furthermore, we are planning to add experiment monitoring in our laboratories, e.g. a system that checks the correctness of the user's actions while conducting specific experiments, and gives advice using natural language. The experimentation process will be more accurately described using additional 2D graphical representations that may display changes of certain physical values through time.

This work has been supported by the Greek Secretariat of Research and Technology under the PENED'99 project entitled "Executable Intensional Languages and Intelligent Multimedia, Hypermedia and Virtual Reality applications", Contract No. 99ED265.

8. REFERENCES

[1] Avradinis, N., LOGOS, European Competition for Educational Multimedia Software of the European Commision, December 1996, finalist, Competition of the Greek Ministry of Education, second prize, January 1997.

[2] Panayiotopoulos T., Avradinis N., Marinagi C.C. "Using Forward Temporal Planning for the Production of Interactive Tutoring Dialogues", Intelligent Systems & Control Conference, EURISCON'98, Athens, June 1998, also in Advances in Intelligent Systems: Concepts, Tools and Applications, (S. Tzafestas ed.), Chapter 20, pp.219-230, Kluwer Academic Publishers, Netherlands, 1999.

[3] Panayotopoulos, A., Panayiotopoulos, T., Avradinis, N., "Logos: A Tutoring Dialogue System for the teaching of Philosophy", In M.H.Hamza (Ed), Proceedings of the 16th IASTED International Conference on Applied Informatics, IASTED-ACTA PRESS, pp. 237-239.

[4] Van Gorp, M. J., Boysen, P., "ClassNet: Managing the Virtual Classroom", In WebNet World Conference of the WWW, Internet & Intranet, 1996.

[5] Krieger, J., "Doing Chemistry in a Virtual World", Chemical & Engineering News, December 9, 35-41,1996.

[6] Dede C., Salzman, M., Loftin, B., "The development of a virtual world for learning Newtonian mechanics". In P. Brusilovsky, N. Streitz (Eds.), Multimedia, Hypermedia and Virtual Reality. Springer Verlag, Berlin, 1996.

[7] Firmeza, J. N. , Ramos, M. S., "Designing a Distance Learning Teleproducts System Supported On The Web", AACE ED-MEDIA World Conference on Educational Multimedia, Hypermedia & Telecommunications, 1998.

[8] Panayiotopoulos, T., Zacharis, N., Vosinakis, S., "Intelligent Guidance in a Virtual University", IMACS International Symposium on Soft Computing in Engineering Applications, SOFTCOM'98, Athens, June 1998, also in Advances in Intelligent Systems: Concepts, Tools and Applications, (S. Tzafestas ed.), Chapter 10, pp.109-118, Kluwer Academic Publishers, Netherlands, 1999.

[9] Panayiotopoulos, T., Katsirelos, G., Vosinakis, S., Kousidou, S., "An Intelligent Agent Framework in VRML worlds", Third European Robotics, Intelligent Systems & Control Conference, EURISCON'98, Athens, June 1998, also in Advances in Intelligent Systems: Concepts, Tools and Applications, (S. Tzafestas ed.), Chapter 3, pp.33-43, Kluwer Academic Publishers, Netherlands, 1999.

[10] Crosby, M. E., Iding, M. K., "The influence of a multimedia physics tutor", Computers & Education, 27 (23), 1997, pp. 127-136.

[11] Roussos, M., Gillingham, M. G., "Evaluation of an Immersive Collaborative Virtual Learning Environment for K-12 Education", presented in AERA Roundtable session at the American Educational Research Association annual meeting, April 1998, San Diego, CA, USA, appears at: http://www.evl.uic.edu/mariar/DOCS/aera_paper.html

[12] Zeltzer, D., "Autonomy, Interaction and Presence", Pres ence, 1, 1992, pp. 127-132.

[13] Witmer, B.G., Singer, M.J., "Measuring Presence in Virtual Environments: A Presence Questionnaire", Presence, 7 (3), 1998, pp. 225-240.

[14] Whitelock, D., Brna, P. and Holland, S., "What is the Value of Virtual Reality for Conceptual Learning? Towards a Theoretical Framework", In Edicoes Colibri, Proceedings of the European Conference on Artificial Intelligence in Education, Lisbon, pp. 136-141, 1996.

[15] Dede, C., "The evolution of constructivist learning environments: Immersion in distributed virtual worlds", Educational Technology, 35 (5), 1995, pp 46-52.

[16] Dede C., Salzman, M., Loftin, B., Sprague, D., "Multisensory Immersion as a Modeling Environment for Learning Complex Scientific Concepts", In Roberts, N. et. al. (Eds.), Computer Modeling and Simulation in Science Education, Springer-Verlag, 1999.

[17] Roussos, M., Johnson, A., Moher, T., Leigh, J., Vasilakis, C., Barnes, C. "Learning and Building Together in an Immersive Virtual World", Presence - Teleoperators and Virtual Environments, 8(3), pp. 247 – 263, 1999.

[18] Prokopenko, M., Jauregui, V., "Reasoning about actions in Virtual Reality", In Proceedings of IJCAI-97 Workshop on Nonmonotonic Reasoning Action and Change, 1997.

[19] Reed, J.A. & Afjeh, A. A., "Developing Interactive Educational Engineering Software for the World Wide Web with Java", *Computers & Education*, 30, 1998, pp. 183-194.

[20] Dede C., Salzman, M., Loftin, B., Ash, K., "Using virtual reality technology to convey abstract scientific concepts", In M. Jacobson & R. Kozma (Eds.), Learning the sciences of the 21st century: Research, design and implementation of advanced technological learning environments. Lawrence Erlbaum, Hillsdale, NJ, (in print).

[21] Arnold, K., Gosling, J., The Java Programming Language, Second Edition, Addison Wesley, New York, 1998.