INTELLIGENT VIRTUAL ENVIRONMENTS FOR TRAINING IN NUCLEAR POWER PLANTS

Gonzalo Méndez, Pilar Herrero, Angélica de Antonio
Computer Science School, Technical University of Madrid
Campus de Montegancedo s/n, 28660 Boadilla del Monte (Madrid), Spain
gonzalo@gordini.ls.fi.upm.es, {pherrero,angelica}@fi.upm.es

Key words: Intelligent Tutoring System, Pedagogical Agent, Virtual Environment, Training, Software Architecture, Nuclear Power Plant.

Abstract: Educational Virtual Environments are gaining popularity as tools to enhance student learning. These environments are often used to allow students to experience situations that would be difficult, costly, or impossible in the physical world. At the Technical University of Madrid we have developed several applications to explore the use of intelligent tutors in VR. In this paper we present two of these applications which have been used for training in radiological protection in Nuclear Power Plants (NPP). These applications are inhabited by avatars and/or agents which are continuously monitoring the state of the environment and manipulating it periodically through virtual motor actions. Our applications help students learn to perform physical, procedural tasks in some different risky areas of NPP.

1 Introduction

Virtual Reality (VR) and Virtual Environments (VE) can provide us with simulation-based learning environments, offering exciting opportunities and challenges for educational software and for intelligent tutors. As in any simulation-based learning environment, students may reach impasses or fail to recognize learning opportunities, so they can benefit from a computer tutor that can provide answers to their questions and offer advice. Currently, there are mainly two kinds of intelligent tutors that are being used together with VEs: Intelligent Tutoring Systems (ITS) (Sleeman and Brown, 1982) and pedagogical agents (Johnson et al., 2000), some of which are discussed in section 5.

VEs offer a broader flexibility for human-computer interaction than earlier technologies did. First, the computer tutor can inhabit the virtual world along with students, which allows a wider variety of interactions between students and tutors. Second, VR allows the tutor to track students' visual attention and physical movements (e.g., the position and orientation of their hands). Thus, VR opens up new possibilities for teaching physical tasks.

VEs are especially valuable in domains where real-life training is very expensive or where students can experience some risky situations, such as maintenance or control of Nuclear Power Plants (NPP) (Pantelidis, 1996). In addition, VR can support more believable stimuli and reactions than earlier technologies, thereby providing an adequate simulation for a wider range of situations.

To explore the use of intelligent tutors in VR, we have developed two applications for training in NPPs: PRVIR (Virtual Reality Technology applied to Training in Radiological Protection) (Mendez et al., 2001) and MAEVIF (Model for the Application of Intelligent Virtual Environments to Education and Training) (de Antonio et al., 2003). Our applications help students learn to perform physical, procedural tasks, such as the procedure for admission in a NPP or the entrance in a radiologically controlled area.

The PRVIR application was developed for training operators of Nuclear Power Plants (NPP) in radiological protection.

PRVIR is divided into two sections. The first one teaches the operators the concepts related to radiology and radiological protection: what radiation is, types of radiation, procedures, etc. It is designed as a multimedia course, where the student has to learn some concepts and pass an exam before he can advance in the course.

In the second section, the student is shown how he must act inside the plant, and then he has to perform inside a VE the procedures he has seen (see Figure 1).

Figure 1: A sample VE.

The ITS monitors the student's actions inside the VE, giving hints when these actions are not correct.

This paper begins describing the ITS we have used in the PRVIR application (section 2) and the architecture for its integration with the VE (section 3); it then presents our current and future work (section 4) and it analyses some related work in the area (section 5); finally, it details some of the conclusions we have obtained (section 6).

2 Structure of the ITS

The ITS we have developed consists of four modules, each of which provides a very precise functionality: tutoring module, expert module, student's module and communication module, as originally described in (Wenger, 1987).

The *Expert Module* contains the knowledge about the subject to be taught to the student, and it is the base for the analysis of the answers provided by the student to the tutor's questions. This knowledge is divided into informative concepts, which are small pieces of information, and the knowledge necessary to solve the exercises. The expert module must be designed in a way that information is easily accessible and modifiable. The way we have overcome this problem is saving all the information in a relational database. Each informative concept points to other concepts that must be shown necessarily before or after it, as they are all part of a particular block of concepts. Each block, in turn, points to other blocks that must be shown necessarily before or after it within a given module.

The *Tutoring Module* contains the pedagogic knowledge, and is in charge of selecting the appropriate concepts to be shown in the course. It also has the strategies, rules and processes needed to drive the interactions between the student and the system, in order to make decisions about the concepts to teach and the exercises to be done by the student, along with the moment when he must be interrupted in order to correct him or make a suggestion. In addition, it decides when it is appropriate to end showing informative concepts and start with an evaluation. At the end of each block of concepts, there is a bunch of exercises related to the concepts explained in that block, so the student can test his recently acquired knowledge. There is also an evaluation at the end of each module, where exercises from all the blocks that form that module will be chosen.

The *Student's Module* keeps individualized information about every student taking the course. It is responsible for tracing what informative concepts have already been taught to the student, how many exercises he has done and the degree of success and time he has used to complete them. To measure the student's progress, we need some metrics with which to compare what are the minimum and average levels for a student to pass to the next level.

The *Communication Module* is in charge of the communication between the student who is taking the course and the ITS. This module must inform the tutoring module about the actions that are performed by the student all along the course. These actions may be the visualization of an informative concept, the answering to an exercise in any of the forms that it may adopt, or any of the actions performed inside the VE. The communication module must make use of all the available multimedia resources in order to make the course as easy-going as possible, but ensuring it does not make the course be too slow, which in the end could bore the student.

Each of the described modules has a very important role to play in the correct operation of the ITS.

3 Architecture

The architecture is the key issue in the correct functioning of the system we have just described. While doing so, we have provided some clues on how this integration may be done. Now, we will explain in detail how this has been done in our system.

From the former section we can infer the following information:

- The communication module must inform the ITS about the actions of the student.
- From the point of view of the tutoring module, there are two different types of exercises. One of them includes the situations simulated in the VE, and the other one encompasses all the different exercises related to the basic concepts.

This means that we can consider the VE as a particular kind of exercise and that the actions performed in it must be supervised by the tutoring module. Thus, the structure of the integrated application is one shown in Figure 2.

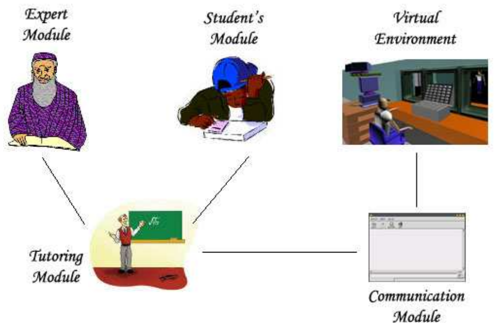

Figure 2: Architecture of the integration.

Once the general structure of the application has been decided, it is necessary to design the way the tutoring module is going to be able to supervise the actions that take place inside the VE.

The steps to be followed inside the NPP are very well defined, and an action cannot be carried out if the ones that go before that one haven't been performed already. Thus, the representation of a procedure using state diagrams, where the actions that make the state change are the actions that the student can do inside the VE, seems to fit perfectly our needs (see Figure 3). In some situations, there may be different possibilities in which several actions can be performed in a not predefined order. In this case, the state diagram may have some branches, but they will eventually converge in a state from where the process will continue.

When these diagrams are designed, the most flexible way to implement the tutoring module is to build a general mechanism that is able to read the structure of any state diagram as well as the actions that must be done to change from one state to another. The state diagram, as we have already done with the rest of the expert module, is stored in a relational database.

The last problem to be solved has to do with the communication between the VE, the communication module and the tutoring module. An action inside the VE might not be possible due to the following two reasons:

- The selected action may not be the right action to be performed at that moment, either because it is not to be done at that point in the sequence of actions or because it is not part of the procedure. This decision must be taken in terms of the expert knowledge.
- It may not be possible to physically do the action, usually because the object is too far to be able to interact with it. This has to do with restrictions inside the VE, and not with the ITS.

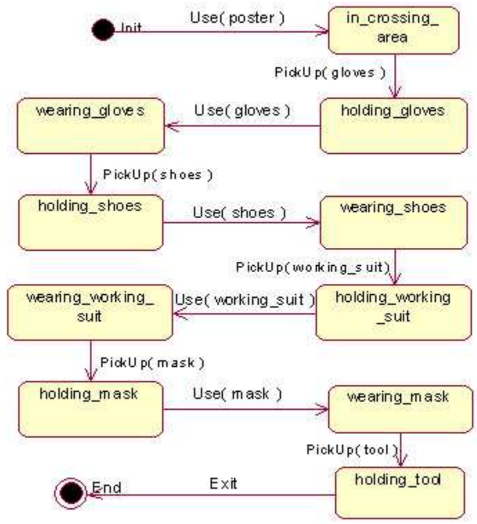

Figure 3: A sample scene.

Because of these two sources of problems, when the student tries to do something inside the VE, the procedure to be followed is:

1. The communication module asks the VE if it is physically possible to do that action.
2. If it is possible, then it asks the tutoring module if it is the right action to be done in that moment.
3. The tutoring module asks the expert module if it is the right action to be done in that moment.
4. If it is the right action, then the tutoring module tells the VE to perform the action.
5. If it is not possible to do the action or if it is not the right action, then the communication module shows a message to the student. This message is sent by the tutoring module in case the action is wrong, or by the VE if it is not physically possible to do that action.

4 Current and Future Work

PRVIR, although successful, has some serious restrictions imposed by our clients at the NPP, and they mainly have to do with the tutor not letting the student deviate from the predefined procedures, so that he can explore other possibilities, or with the adaptation of the tutor to the student's needs.

Thus, the tutor in this application is not very intelligent or skilled, and, as an additional drawback, it is not embodied, so it cannot demonstrate the student how to perform the procedure he is studying.

Even so, the PRVIR system is already in use at the Vandellos II NPP in Spain. Despite the limitations we have mentioned, this has been the first step to use VR in a Spanish NPP to train operators in different tasks, so depending on how successful it is, we will be adding more capabilities to the intelligent tutor, so that in a close future these system can effectively be used instead of the training inside the NPP.

Several experiments have shown that the learning experience is much more effective if there is an embodied tutor inside the VE helping the student in the training process (Lester et al., 1997a). Because of this reason, we are currently developing a new system, MAEVIF, where we are substituting PRVIR's ITS with a multiagent system that will have analogous functions, but that will support an embodied tutor and will also be easier to expand and modify.

MAEVIF (Model for the Application of Intelligent Virtual Environments to Education and Training) is a project funded by the Spanish Ministry of Science and Technology. The objective of the project is the definition of a model for the application of intelligent virtual environments in education and training. This implies:

- The definition of a generic model for intelligent learning environments based on the use of virtual reality.
- The definition of an open and flexible software architecture to support the generic model of a learning environment.
- The design and implementation of a prototype authoring tool that simplifies the development of learning environments based on the generic model.
- The definition of a set of methodological recommendations for the development of virtual learning environments. This methodology will propose a set of steps for the design of the environment according to the generic model, and its implementation with the help of the authoring tool.

The most relevant aspects that are being taken into consideration in the definition of the generic model are:

- The design of a tutoring strategy, easily adaptable and configurable, that allows for an intelligent supervision as well as for the dynamic adaptation of the system to the special needs of each student.
- The design of a cognitive diagnosis method which is able to perform non-monotonic reasoning about the knowledge of the student and their individual characteristics.
- The definition of a knowledge representation formalism for the expert knowledge which is specifically crafted for the use within virtual environments, and which is independent of the language used for its development.

Each component of the architecture of the intelligent tutor is an agent with a very specific function, the most relevant of which are:

- Planning agent: it uses a strips-based algorithm to plan and replan what the students have to do. This makes it possible to dynamically generate exercises, so that students can practice different procedures starting from different situations.
- Tutor agent: according to the planning agent's plan, this agent is in charge of explaining the student what he has to do, giving hints, supervising him or showing the appropriate actions.
- Plath-planning agent: this agent is in charge of calculating the appropriate routes to go from one place to another inside the VE using the RTA* algorithm. It also controls the student's movement to see if he is following the right path. If not, it gives hints to the student so that he can follow the right route.
- World agent: its mission is to keep a snapshot of the state of the world, so that he has all the information about the world that other agents may need. This information includes the position of all the objects and avatars, the state of different devices or the objects that each student is carrying.

The communication among agents is carried out using a blackboard. Every time an agent needs something, it publishes it on the blackboard. The rest of the agents will see if they have something useful, and will write it on the blackboard, too. The original agent will then see all the options and choose the one that best suits its needs.

This mechanism has been used, for example, by the planning agent. This agent publishes the state of the plan, and asks the rest of the agents if they can contribute something to the plan. With all the answers, the planning agent explores the different alternatives until the plan is finished.

MAEVIF supports several students being trained at the same time using different computers connected though a network.

The agents are being developed using JADE, and the VE has been developed using OpenGL. The connection between both of them is based on CORBA, so, as an additional feature, they can run in different machines to improve the system's performance. Finally, the connection between different clients is done using DirectPlay, so the visual state of the VE is consistent in all the clients.

5 Related Work

There are several projects aiming at the use of VR for education and training.

In all probability, the most famous embodied pedagogical agent nowadays is Steve(Soar Training Expert for Virtual Environments). Steve is an autonomous, animated agent for training in 3D virtual environments (Rickel and Johnson, 1999) that has been developed at USC. Steve's role is to help students learn procedural tasks, and he has many pedagogical capabilities one would expect of an intelligent tutoring system: Steve can demonstrate procedures, it can monitor students as they practice a task, giving them feedback on their actions, and it can answer simple questions.

Steve was designed to be easy to use in new domains and virtual worlds. It was originally applied to equipment operation and maintenance training on board a virtual ship. Subsequently, it was significantly extended and applied to leadership training in virtual Bosnia (Rickel et al., 2002).

However, the leadership training application was designed with Steve in mind. We have recently carried out an experiment to integrate Steve in an already existing VE (Mendez et al., 2003), and although it worked quite well, some issues did arise that require further consideration in the design of this kind of agent.

Adele (Agent for Distance Education - Light Edition) (Shaw et al., 1999) has also been developed at USC. The functionality of Adele is quite similar to that of Steve, but it has been extended to support some additional persona features and instructional capabilities that Steve lacks. In addition, whereas Steve has been thought to be used for training, Adele has been mainly designed to be used in education.

Adele has been used in a case-based clinical diagnosis application, and she can highlight interesting aspects of the case, monitor and give feedback as the student works through a case, provide hints or rationales for particular actions, or quiz the student to make sure he understands the principles behind the case.

Cosmo is a life-like animated agent developed at NCSU IntelliMedia Initiative (Lester et al., 1997b). Given a request for an explanation or a hint, Cosmo's behavior planner selects the explanation, which is mainly determined by the current problem state. Then, the explanation planner consults the knowledge sources to select a sequence of communicative acts. To ease the student's acquisition of problem-solving skills, the explanation planner supplies him with relevant causality knowledge when the student requests advice, as well as justifications for its suggestions.

A lot of effort has been spent to endow this agent with diectic believability, which allows him to move through the environment, point to objects and refer to them appropriately. For that, a diectic behavior planner is used to coordinate locomotive, gestural and speech behaviors.

Herman the Bug is another life-like pedagogical agent developed at NCSU IntelliMedia Initiative (Lester et al., 1999) that has been used in the DESIGN-A-PLANT learning environment. The objective is for students to learn concepts about botanical anatomy and physiology.

Herman is an insect that dives into plant structures and provides problem-solving advice to students. As students build plants, Herman observes their actions and provides explanations and hints.

This learning environment was built in order to study mixed-initiative problem-solving interactions in constructivist learning environments. Herman cannot participate in complex dialogues requiring turntaking, back channeling, or even rudimentary discourse segmentation. The authors have been able to identify what they have called the *persona effect* (Lester et al., 1997a), meaning that the presence of a lifelike character in an interactive learning environment - even one that is not expressive - can have a strong positive effect on student's perception of their learning experience.

Vincent is a synthetic pedagogical agent that helps the trainees in the web based learning process (Paiva and Machado, 2002). This agent combines a set of sensors and actors that establish message-based communication with the micro-learning environments while gathering information about trainee performance. It has an anthropomorphic representation featuring four emotional attitudes: sad, happy, disappointed, and impatient.

Vincent can perceive the environment and act on it, having the capability of making inferences about those perceptions, solving problems and determining what actions should be performed to reach his goals. He can do this through his cognitive behavior, which implies deciding what pedagogical actions must be taken for a particular situation, and his physical behavior, which includes Vincent's visual and audio attitudes.

The use of pedagogical agents seems to be more extended than ITSs, due to the benefits obtained from their embodiment inside VEs, and this tendency will increase when these agents are commonly endowed with human attributes such as perception, personality traits, natural language recognition and generation or ergonomical restrictions, since training will be much more realistic.

6 Conclusions

Although secure, Nuclear Power Plants are a very special environment where all precautions and help are always welcome. As we learned during the development of the PRVIR project, there are certain areas where human presence is not advisable. However, from time to time it is necessary to inspect their state, and in these cases, the better the action is planned and trained, the less dangerous it is for the person who has to perform it.

For these reasons, NPPs will be really benefited from the advance in the development of VEs as a substitute for physical mockups of the plant, since they constitute a more economical solution for planning and training than former ways do.

Our first experience adding intelligent tutoring in the PRVIR project has been quite satisfactory, and although the system had some limitations, it provided us and the Vandellos NPP with a very valuable experience to carry on with this work.

As a result of this experience and our previous work with ITSs and agents, we expect the MAEVIF system to be a much more sophisticated substitute for VEs for training with intelligent tutoring, in which it will be possible to substitute any of its components with a different one in order to better adapt the system to the particularities of each domain and user, as well as to take advantage of new advances in science and technology.

For this to be possible, it would be desirable that all the researchers and developers of this kind of systems worked towards the elaboration of standards that allowed the construction of interchangeable components. These standards will be quite beneficial for the development of VEs for training, due to the wide range of disciplines involved in the development of these systems and the difficulty to have experts in all of them in the development teams.

Unfortunately, as far as we can see, these standards are still far from being available.

7 Aknowledgements

PRVIR was funded by the Electrical Group for the Nuclear Technological Development (DTN). MAEVIF is funded by the Spanish Ministry of Science and Technology under contract TIC2000-1346.

REFERENCES

de Antonio, A., Imbert, R., Ramirez, J., and Mendez, G. (2003). An agent-based architecture for the development of intelligent virtual training environments. In *Proceedings of m-ICTE 2003*.

Johnson, W. L., Rickel, J. W., and Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11:47–78.

Lester, J., Convers, S., Kahler, S., Barlow, T., Stone, B., and Bhogal, R. (1997a). The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI 97*, pages 359–366. ACM.

Lester, J., Voerman, J., Towns, S., and Callaway, C. (1997b). Cosmo: A life-like animated pedagogical agent with deictic believability. In *IJCAI97 Workshop on Animated Interface Agents: Making them Intelligent*, Nagoya, Japan.

Lester, J. C., Stone, B. A., and Stelling, G. D. (1999). Life-like pedagogical agents for mixed-initiative problem solving in constructivist learning environments. *User Modeling and User-Adapted Interaction*, 9(1-2):1–44.

Mendez, G., de Antonio, A., and Herrero, P. (2001). Prvir: An integration between an intelligent tutoring system and a virtual environment. In *SCI2001*, volume VIII, pages 175–180, Orlando, FL. IIIS, IEEE Computer Society.

Mendez, G., Rickel, J., and de Antonio, A. (2003). Steve meets jack: the integration of an intelligent tutor and a virtual environment with planning capabilities. In *Intelligent Virtual Agents. 4th International Workshop, IVA03*, number 2792 in LNAI, pages 325–332. Springer-Verlag.

Paiva, A. and Machado, I. (2002). Life-long training with vincent, a web-based pedagogical agent. *International Journal of Continuing Engineering Education and Life-Long Learning*, 12(1).

Pantelidis, V. S. (1996). Suggestions on when to use and when not to use virtual reality in education. *VR in the Schools*, 2(1):18.

Rickel, J. and Johnson, W. L. (1999). Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence*, 13:343–382.

Rickel, J., Marsella, S., Gratch, J., Hill, R., Traum, D., and Swartout, W. (2002). Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, 17(4):32–38.

Shaw, E., Johnson, W., and Ganeshan, R. (1999). Pedagogical agents on the web. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 283–290, Seattle, WA, USA. ACM Press.

Sleeman, D. and Brown, J., editors (1982). *Intelligent Tutoring Systems*. Academic Press, London.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann Publishers.