Efficient Architecture for Low-Cost Adaptive 3D Graphics-Based E-Learning Applications

Luis Salgado[♂], Jesús Bescós[♀], Francisco Morán[♂], Julián Cabrera[♂], Enrique Estalayo[♂] and José Cubero[♂]

Grupo de Tratamiento de Imágenes

[♂]Universidad Politécnica de Madrid, [♀]Universidad Autónoma de Madrid

Abstract

This work describes the design and implementation of the Virtual Reality Tool (VRT), a core component within the educational platform developed for the European research project IST-2001-33184, SLIM-VRT (Self Learning Integrated Methodology – Virtual Reality Tool). The main objective of this project is to propose an alternative for integrated maritime self-learning on board and ashore based on the use of interactive multimedia and low-cost 3D graphics technologies.

The integration of the Virtual Reality Tool in the educational platform involves the incorporation of 3D graphic experiences to the educational process. Based on VRML and other standard tools, the VRT-enabled case studies are used for those learning activities in which the students interact with a virtual environment either to access multimedia information associated to relevant objects/devices in the scene, or to manipulate those devices to carry out specific simulations of operational tasks identified in the case study.

With the aim of fulfilling the user's requirements and learning needs identified within this field of application, the design of the VRT has required specific developments to overcome VRML limitations and make it possible to: (i) customize dynamically both the education programme and the multimedia content according to the specific user's and terminal profiles; (ii) manage efficiently the interactivity with the virtual environment to adapt in real-time the learning process, according to student's actions; and (iii) design and implement multiple case studies based on the same virtual environments, thus minimizing the modelling effort.

Categories and Subject Descriptors (according to ACM CCS): K.3.1 [**Computers and Education**]: Computer Uses in Education. C.2.4 [**Computer-Communication Networks**]: Distributed Systems. H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems.

1. Introduction

Organizations and working environments are changing rapidly and it is therefore critical for companies to renew their strategies and knowledge to avoid becoming outdated. In this framework, the concept of continuous learning has become prominent since it places priority on noticing, adapting and learning from these changes.

Empirical studies of professional practice show that, while the focus is primarily on getting the job done, learning is inextricably intertwined with working. In this sense, professionals must continually learn to apply existing knowledge to routine, innovative or emergency situations, and to construct new knowledge in response to changing workplace situations. Employees must keep up with the speed of technological, market-driven and organizational changes, so the need for acquisition of new and broader competencies grows, and the capability and willingness to learn becomes a prerequisite for the modern worker. Nevertheless, and despite its importance, there are specific sectors in the society for which satisfying this prerequisite becomes difficult, as they have to deal with limitations regarding the mobility of the trainees or the cost associated to the time they spend out of their working place. This is the case, for example, of the maritime industry, in which employees stay on board most of their working lifetime.

Currently, the shipping sector relies on traditional "school models" for most of the training needs, even though workplace learning is fundamentally different from traditional school learning. In this traditional approach, there is a need to promote the new arising practices: workers typically receive a pre-designed course in the new regulations, procedures or processes –often at a different location (i.e., ashore) from their place of work (i.e., on board)– and are expected to apply this abstracted knowledge later in their workplaces, sometimes with disappointing results.

To overcome this situation, new innovative approaches must appear at the levels of context learning, pedagogical methodology, technology applications and evaluation procedures. Therefore, the rising strategies should provide:

- Movement from one single passive training course to continuous interactive learning processes.
- Dynamic generation of education programs and contexts according to the individual user needs.
- Support of user friendly learning methods, knowledge sharing, mentoring, learner-tutor communication, etc.

One approach which satisfies all these requirements is the e-learning methodology. E-learning can be defined as the effective learning process created by combining digitally delivered content with learning support and services [WW01]. Thanks to the very nature of its digital content, one of the technologies that has fostered the most the spread of this new learning process has been the Internet. The "World Wide Web" provides many communication capabilities allowing the students to have an ubiquitous access to the contents of the e-learning applications.

Furthermore, the increasing multimedia capabilities of most common web browsers have enabled the incorporation of 3D graphics as a new accessible media for the general public. Web browsers can interpret and render interactive 3D scenes described in standard languages –such as VRML [HW96][ISO97] or X3D [ISO04]– thanks to publicly available plug-ins, thus enabling the generation of low-cost 3D graphics which can be used as additional learning material. The virtual worlds, generated according to real scenarios, are presented to the students, who experience the scene in a more interactive and immersive manner. Hence, the virtual experience improves the learning process by providing new educational capabilities.

The integration of the 3D graphics within the e-learning methodology, overcoming the limitations related to the learning and training in the maritime sector, is the framework where the European project IST-2001-33184 SLIM-VRT (Self Learning Integrated Methodology – Virtual Reality Tool) [Sli01] was developed. The main objective of this research project was to achieve the integration of self learning on board and ashore using interactive multimedia and 3D graphics.

From the technical point of view, SLIM-VRT proposes the development of an e-learning system architecture with different collaborative modules in which the most relevant and innovative contribution is the incorporation of 3D graphics to the educational process, providing new (active and reactive) interaction capabilities. This contribution is achieved through the design and implementation of a scalable, flexible and efficient tool, the Virtual Reality Tool (VRT), which supports new education methodologies on the basis of case studies. VRT-enabled case studies are used for those learning activities in which the students clearly benefit from interacting with a virtual environment either to access multimedia information associated to relevant objects/devices in the scene, or to manipulate them for carrying out specific simulations of operational tasks identified in the case study.

In this paper, a detailed functional description of the VRT architecture is presented, addressing the main technical achievements, and introducing examples (case studies) to show how the different functionalities support the incorporation of 3D graphics to the learning process. The paper is structured as follows: Section 2 introduces the main requirements driving the design and implementation of the VRT architecture. Section 3 describes the VRT overall architecture, and Sections 4 and 5 deepen into its two main functional modules. The use of 3D graphics in the context of maritime applications is described in Section 6 through the introduction of VRT-based case studies and, finally, some conclusions are listed in Section 7.

2. VRT Requirements and Functionalities

The user's and e-learning needs analyzed in the first steps of the project pointed out a set of general requirements and specific functionalities driving the design and implementation of the VRT architecture, which had to support efficient low-cost 3D graphics-based experiences with:

- Dynamic education programme and multimedia content customization according to user characteristics and terminal technical capabilities. While the student is moving freely within the virtual environment in a case study, changes in his interests, knowledge level or communication characteristics imply the need to adapt the contents to the new session parameters. This fact demands an external update of the different established sessions, in which the system must readapt the contents and interaction mechanisms enabled for each client in an efficient, user friendly and flexible way.
- Efficient management of the interactivity allowing dynamic adaptation of the learning process in response to the student's actions in the virtual environment. When immersed in a virtual world, a student experiences a scene which follows a behaviour pattern established by the objectives imposed in the learning process. This pattern is implemented via an interaction logic that controls the interaction flow between the student and the virtual world and, therefore, must be programmed and adapted for every learning situation.
- Reusability of graphical models to allow the design of multiple case studies based on the same modelled virtual environments.

Low-cost 3D graphics and VR experiences are now possible with publicly available plug-ins which can interpret and render interactive 3D scenes described by using standard languages and tools. Based on these standard tools, the VRT supports the following specific basic functionalities:

- The visualization of 3D objects integrated in complex scenarios according to the topics of the specific courses. The aim is to be able to integrate graphical material into the overall course and make it accessible to the user via a WWW browser.
- A first level of interaction with the 3D objects through standard Input/Output (IO) devices. For example, to allow the inspection of a relevant model via mouse, or to visualize the scenario with a standard monitor.
- Navigation facilities inside the virtual scenarios to allow the students to move freely around the 3D scenes.

In addition to the basic functionalities, the VRT also provides a set of enhanced features which allow developing enriched 3D scenarios in terms of new metaphors for user interaction, information contained within 3D scenarios, and the consideration of immersiveness. In particular, these enhanced functionalities consist of:

- The possibility to define guided paths or tours inside the virtual scenarios so that the user needs not deal with navigation controls. Guided paths are configurable, allow different levels of interactivity, and provide an excellent starting point to present 3D environments to the user.

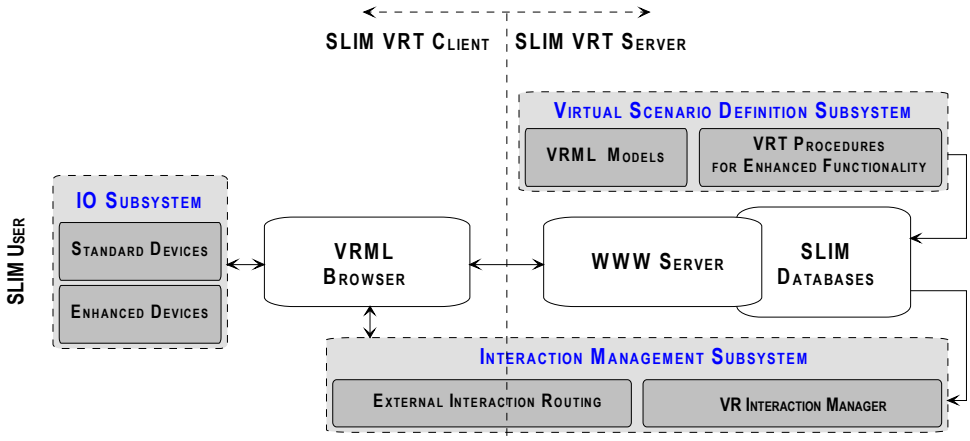

Figure 1: *General block diagram of the VRT architecture*

- Mechanisms to attach multimedia explanations (in the form of video or audio clips, text, WWW links, etc.) to objects and/or locations integrated with the 3D scenario and in a user-friendly way. Furthermore, the user can access this multimedia information interacting directly with the models included in the 3D world.
- Functionality adaptation to user and terminal profiles. The VRT supports the usage of profile information about user's characteristics, such as educational level or language, as well as terminal capabilities (hardware acceleration available, immersive devices, bandwidth, etc.).
- The ability to evaluate user's interaction, in terms of both his location in the virtual scenario (motion vs. timing) and his actions over specific objects. Tracking user's evolution within the virtual scenario is possible, and the corresponding data are available at the server site for further processing: evaluation, monitoring, etc.
- Support of immersive capabilities via the use of enhanced IO devices, e.g., 3D glasses, head and hand trackers, etc. Immersive devices allow the user to enjoy a more realistic experience and interaction with the scene.

3. VRT General Architecture

Figure 1 presents a general block diagram of the proposed VRT architecture for e-learning applications designed to fulfil the previously introduced requirements and functionalities. The VRT was built on top of a standard framework: the VRML standard in combination with a VRML browser, remotely fed by a WWW server, which should support the execution of external JAVA code. This is a main requirement to tailor user interaction with an externally controlled educational system, as well as the DirectX rendering primitives, which directly include further support for stereovision with the most popular 3D graphics cards. The VRT is composed by three specific subsystems, closely integrated into this standard framework:

a. **Input/Output Subsystem:** This subsystem is in charge of providing the user with the appropriate IO services according to the capabilities of the hardware available for rendering the 3D graphics. Two different categories can be distinguished: standard IO, including basic interaction and presentation devices (keyboard, mouse or monitor) directly managed by the standard framework adopted; and enhanced IO, involving the use of special hardware to achieve immersiveness (stereo glasses, trackers, head-mounted displays, etc.) and thus requiring to develop specific software to connect those special devices with the VRML browser in an efficient and portable way. In parallel, the IO subsystem deals also with the metaphors required to achieve user immersiveness.

b. **Interaction Management Subsystem:** This subsystem, fully described in Section 4, aims to establish a bidirectional communication link between the virtual scenario rendered at the client and an *external entity* at the server specifically designed to track and control the user interaction in the scene: the VR Interaction Manager (VRIM). This link is not considered by the standard VRML framework, but is very useful to manage the delivery of user motion and actions from the virtual scenario to the VRIM, which can consider them for evaluation purposes. This link also manages the delivery of controlling events from the VRIM to the virtual scenario, as a way to modify the virtual scenario according to user's actions. Our approach decouples the behaviour pattern of the scene (i.e., the interaction logic) from its geometric description, hence reinforcing both the reusability of the models, which may be linked to different logics, and the flexibility in the design, as parallel developments of the geometry and the logic can be carried out almost independently.

c. **Virtual Scenario Definition Subsystem:** This subsystem, to which Section 5 is devoted, is in charge of providing the graphical models for the generation of the 3D scene which will allow the virtual experience of the student. Furthermore, this subsystem also provides the mechanisms for integrating most of the enhanced functionalities proposed within the project, such as the predefined tours or the association of multimedia educational information to the graphical models, some of them not supported by the standard VRML framework.

4. Interaction Management Subsystem

Some 3D modelling languages, like VRML, allow to code the interaction logic of the scene in the same file where its

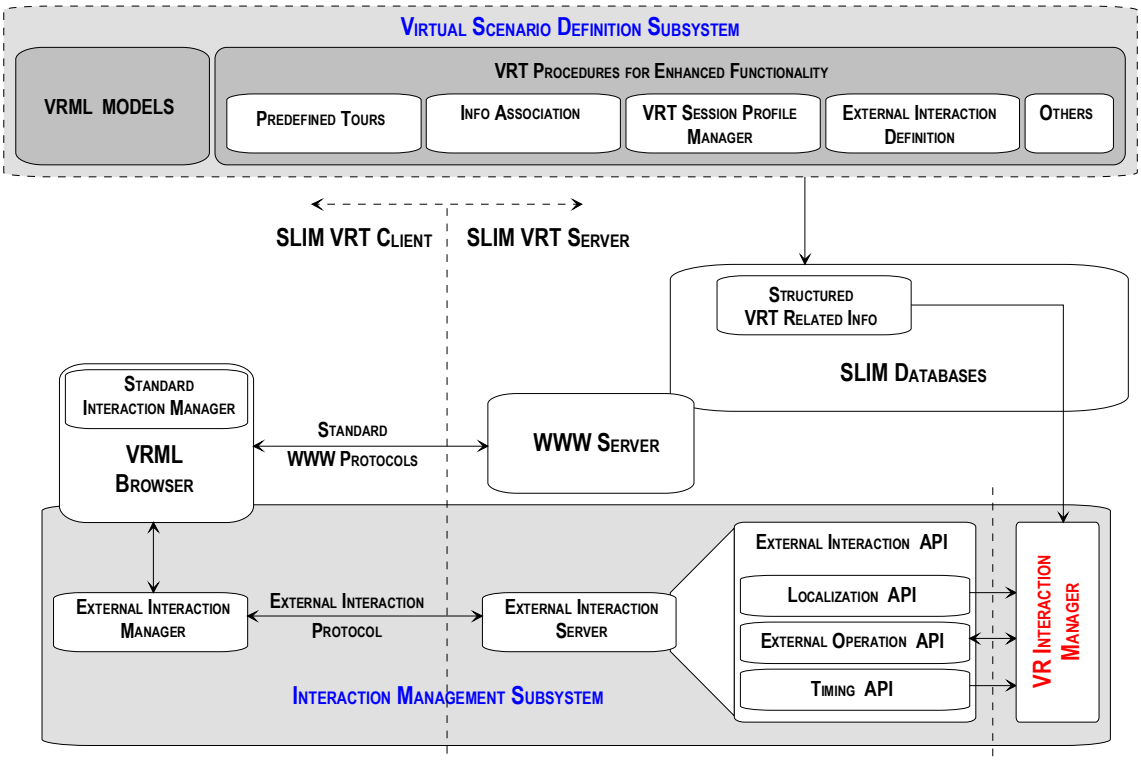

Figure 2: *Detailed block diagram of the IM (bottom) and VSD (top) subsystems*

geometry is defined. The interaction management is usually event-based, an event being just some kind of basic information message sent or received by any scene object or node in an asynchronous way: active nodes (like sensors) defined in the scene launch events that are routed to other active nodes as a basic means to modify their characteristics (e.g., their position). To simulate more complex interactive situations, VRML considers the possibility of defining intermediate script nodes which accept input events, perform some processing according to the received information, and generate consequent output events, which are then routed to the desired nodes. These routing and scripting mechanisms, which conform the interaction logic, generate the traffic of events [WRO04] that contains all the information related to user-scene interaction. In our case, this information flow is established between the student and the browser that interprets and renders the modelled scene.

This scheme results in a lack of flexibility, from at least two points of view. First, every desired modification of the behaviour pattern which controls the learning process implies the reprogramming of the interaction logic embedded in the same file (structure) as the geometric description of the scene. Second, the student's actions cannot be inspected on-line by an external agent (e.g., an evaluator) and the student's perception of the scene cannot be modified on-line according to the criteria of an external agent (e.g., an instructor). These problems prevent any kind of dynamic adaptation of the learning process according to student's actions, which is one of the main features that e-learning can provide over traditional educational paradigms.

In order to solve the first problem, some works propose to separate the interaction logic from the local machine where the scene is rendered [BPT00]. We propose to deepen into this approach and solve all the identified problems by including a client-server scheme not just intended to centralize course management (an architecture frequently used in e-learning systems [WRO04][BT04][CBR*02]) but specifically targeted at the management of the interaction information flow. This innovative approach completely decouples the behaviour pattern of the scene from its geometric description.

Moreover, this approach indirectly achieves a high degree of content protection. Although the scene geometry has to be available in the client for rendering, it is mostly useless (at least, from an educational viewpoint) without the behaviour pattern, which is kept in the server. Hence, the overall scene is inherently protected against undesired copies or misuse. Additionally, our approach enables to achieve the simultaneous on-line inspection and adaptation of the interaction flow for several students, each with his own virtual learning environment (i.e., many clients), from a single evaluation and instruction centre (i.e., one server).

Figure 2-bottom presents a detailed block diagram of the proposed Interaction Management (IM) subsystem architecture. It allows user-scene interactions to be managed by an *external entity,* called Virtual Reality Interaction Manager (VRIM), which implements the scene interaction logic. It runs on a remote server and operates in real time.

The IM subsystem implements a bidirectional event-based communication channel between the virtual scene and the VRIM, on top of a TCP/IP link. It involves several modules and interfaces: the External Interaction Manager (EIM) at the client side, and the External Interaction Server (EIS), together with its External Interaction API (EI API),

at the server side. The EIM first handles the exchange of VRML events with the rendered scene and then establishes with the EIS a TCP connection that follows the External Interaction Protocol (EIP). The VRML events are directed from/to the EIM to/from the EIS, which further communicates with the VRIM via the EI API. The remainder of this Section gives a functional description of these interfaces, modules and protocols (see Figure 2-bottom).

4.1. External Interaction Manager

The EIM is a script that orchestrates the communication with the EIS in order to exchange events between the VRML browser, which renders the scene at the client side, and the VRIM running at the server side.

When the scene is rendered, the EIM establishes a TCP communication link with the EIS. Every event routed from the scene to the EIM is processed by this module and sent to the EIS according to the EIP.

4.2. External Interaction Server

The main function of the EIS is to act as a TCP server accepting connections from multiple clients, and holding bidirectional communications of events with all of them according to the EIP.

4.3. External Interaction Protocol

The EIP regulates the communication of asynchronous messages between the EIM and the EIS. This allows the effective transmission of all types of events produced as a result of the user interaction within the virtual scene.

4.4. External Interaction Events

To achieve a simple and efficient management of the interaction flow for educational purposes, the interaction-related information must be first classified and constrained to a reduced set of event types: the External Interaction Events. These are abstract data types that serialize the information provided by the VRML events and conform the basis of the EIP. There are three kinds of such events, encapsulating different kinds of interaction-related information:

a. Localization Events, indicating the position of the avatar.

b. Timing Events, aimed at transmitting temporal information of the client where the scene is being rendered.

c. External Operation Events, which group both the different VRML output events generated as a result of the user interaction with specific objects in the scene, and the input events that will be generated by the VRIM as a response of the interaction logic to the user's actions.

The first two types are intended to be used for spatiotemporal tracking, e.g., to evaluate the user's position (within the scene) or connection time (for how long the user is experiencing the scene), or how much time the user spends to perform specific required actions.

The last type of events intends to track the user's interaction with specific objects in the scene, and to route the corresponding responses provided by the VRIM which might modify the scene appearance according to the specific implementation of the behaviour pattern.

4.5. External Interaction API

The EI API is the interface between the EIS and the VRIM. This interface is the one that effectively decouples the behaviour pattern of the scene from its geometric description. It separates the fixed framework, which allows to control the interaction externally, from the potentially many implementations of the VRIM, which finally define the interaction logic. The EIS functionality (i.e., TCP communication management) is completely independent from the particular virtual scene, whereas the VRIM implementation fully depends on it, as VRIM manages the user's actions and the corresponding scene reactions. Therefore, the EI API allows a flexible, efficient and updatable implementation of the interaction management.

The EI API is subdivided in three APIs, each managing the communication of a type of event: there is a Localization API for the Localization Events, a Timing API for the Timing Events and an External Operation API for the External Operation Events.

4.6. Virtual Reality Interaction Manager

The Virtual Reality Interaction Manager (VRIM) implements the behaviour pattern of the scene. It receives events related to the user-scene interactions through the EIM, EIS and EI API, and sends back events in response and through the reverse path. The VRIM is also able to self generate messages (events) to be sent to the user at any time, e.g., upon the orders of an instructor.

The functionality offered by the EI API allows a single VRIM implementation to simultaneously handle interaction with multiple clients, which opens the way for interaction between clients.

5. Virtual Scenario Definition Subsystem

The objective of the Virtual Scenario Definition (VSD) subsystem is the generation of an adaptive environment that incorporates different procedures for enhanced functionalities which can help the student during the educational process. These enhanced procedures are detailed below (see Figure 2-top):

5.1. Predefined Tours

In many occasions, the information available within the virtual scene can be ignored or difficult to access, either due to the environment complexity or to the lack of user's experience. In these situations, the inclusion of predefined tours is a very useful mechanism to show the overall virtual scene, also focusing the attention of the avatar on the most relevant aspects of the application.

The implementation of these predefined tours is not standardized in VRML. This is why the VSD subsystem integrates templates allowing the content designer to specify the path that the user will follow. These templates have been used during the generation of the different case studies. Moreover, this functionality has been combined with the integration of 2D maps inside the virtual scene, which allows the student to know his position within the scenario, even when he is following a predefined tour.

5.2. Information Association

The 3D scenes provide themselves a great amount of information as they allow the user to navigate through different environments and, sometimes, to interact with the elements placed inside. Nevertheless, in many educational situations, it may be considered that this information is not enough, or that it could be further complemented via associating external multimedia data in order to improve the level of knowledge supplied to the student.

The association of this new information to the virtual scene is not straightforward. The external files containing the new contents need to be instanced inside the VRML files and some kind of new interface has to be generated for accessing these external files. In the VSD subsystem, this interface has been implemented by enclosing new VRML models within the scenario description, and following a three steps interactivity model, which is shown in Figure 3:

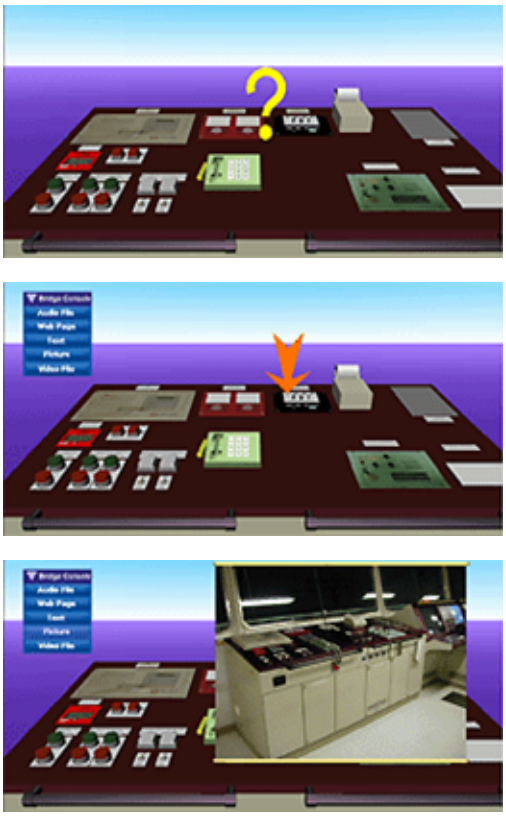

Figure 3: *Interface for accessing the associated multimedia information inside the virtual scenario*

a. First, a question mark is displayed when the avatar gets close enough to any of the relevant scene objects having additional associated information (Figure 3-top).

b. If the user clicks on the question mark, a textual menu is rendered with details on the information and formats accessible for the model at that time (Figure 3-middle).

c. Finally, if the user selects one of the different menu options, the information related to it is displayed in the virtual scenario, integrated within an unrolled scroll (e.g., a picture of the real object, as in Figure 3-bottom).

5.3. VRT Session Profile Manager

The e-learning platform provides service to many different clients, which can be grouped into profiles according to different parameters. These profiles are used in the VSD subsystem to customize the contents of the VRT (both the VRML models and the information associated to them) so that they can be adapted for optimizing not just the rendering of the scene, but also the learning experience of the students.

The parameters taken into account in these session profiles are:

- User's language: the VRT can select among different language versions of the multimedia materials associated to specific 3D models.
- Educational user's level: the VRT is able to determine which associated materials have to be presented to the student according to his grade.
- Hardware capabilities: the VRT can configure 3D scenes depending on the characteristics of the user's terminal: hardware acceleration, capability of stereo vision, 6DOF pointing devices, etc.

5.4. External Interaction Definition

The user interaction, which is managed outside of the virtual scene, has to be defined according to the VRML standard mechanisms: it is necessary to specify which objects will be subject to the external interaction, and which type of information they will provide or accept. This specification, the set of rules and labels that should be added to the geometric description of the scene, is identified as the External Interaction Definition. It consists basically of the list of the VRML events from the scene objects allowing external interaction management, and is used to communicate the VRML Browser and the EIM (see Figure 2-bottom).

5.5. Dynamic (Un)Loading of the Virtual Scenarios

The modelled environment within the SLIM-VRT project has very large dimensions which would require a long time and a great computational effort for its rendering in case it is presented completely to the client. The solution implemented in the system to overcome this limitation is to download and render in the browser only those parts of the model which are or can be visualized by the user: when the user changes from one location to another, the previous part

of the scene is unloaded transparently to the student, and the models corresponding to the new part are rendered.

The dynamic process of (un)loading elements inside a virtual scenario is not defined by the VRML standard. We have thus developed a new module which controls the elements of the scenario to be added or removed from the scene depending on the avatar's position and on the student's interactions (e.g., if a door is opened, the furniture of the room is loaded; and if it is closed, the room is emptied).

5.6. Dynamic Adaptation of the Information

The VRML worlds publicly available have so far been relatively limited in terms of scalability and adaptability: changes in the user's interests or knowledge level, or in the communication channel characteristics, did not imply the adaptation of the contents to the new situation, as the functionalities provided by the VRML standard do not support the updating of the scene contents according to any profile.

A possible solution for this shortcoming is either to modify the description of the scene or to generate different versions of it, one for each supported profile. Then, whenever the session parameters change, the VRML code has to be re-downloaded, replaced and rendered at the client to achieve the desired adaptation. But such a monolithic approach implies severe limitations: long waiting intervals; pausing of the visualization process at the client site due to the downloading and rendering of a new instance of the virtual world; and, additionally, the loss of the history of activities carried out by the student up to that moment.

In order to partially overcome the VRML limitations in adapting content to the parameters of a user session, we have fully decoupled the multimedia information associated to the models (see Section 5.2) from the VRML description of the scenarios. The new strategy consists in defining an external logic that is in charge of dynamically managing the association and adaptation (from now on: associative and adaptive processes) of the educational multimedia content linked to the virtual scenarios. This approach further reinforces the reusability of the 3D models (since they do not have to be readapted to different content or content updates) as well as the flexibility in the design.

This dynamic management of the associated information is carried out in the following way:

- A database stores meta-information of the educational multimedia content associated to the 3D models inside the virtual scenario. This meta-information, coded in XML, gathers data such as the location of the different files inside the VSD subsystem or to which 3D model these files are referred.
- An external logic, implemented in Java, is in charge of adapting the associated information, a process which takes place each time a student requires additional information associated to one of the models placed inside the virtual scene (i.e., when he clicks on the model of a question mark displayed above the 3D model which contains associated information: see Figure 3-top). The steps followed by the adaptation process are:

  a. When the external logic receives a request asking for multimedia information associated to a specific model from the student's terminal, it accesses the VRT session profile for that specific client and gathers the session parameters.

  b. Next, it combines the acquired session parameters with the meta-information of the multimedia contents, and determines which specific information is accessible for that session and object.

  c. The availability of the different contents associated to that model is communicated to the student through the presentation of a textual menu (Figure 3-middle).

  d. Finally, when the student picks a menu option, the information selected is downloaded from the server into the client and rendered in the virtual scenario (Figure 3-bottom).

6. Case Study

The features intended to complement a learning process with interactive 3D graphics have been integrated and tested via several case studies developed in the field of maritime education. Case studies offer learning contents with increasing level of complexity in terms of interactivity and immersiveness, from simple static 3D models for passive visual inspection to highly immersive environments with rich interaction possibilities. Case studies demonstrate that the SLIM-VRT system can be used as a low-cost simulator, with adaptable and scalable content (3D graphics, and multimedia information associated to them), customizable environment behaviour, and online inspection and logging capabilities. Therefore, the actions carried out by the user in the virtual environment, as part of the exercises proposed in the case studies, can be remotely monitored by tutors or supervisors for assessment and evaluation.

Case studies are basically exercises to complement topics of specific courses where the student has to put in practice concepts and protocols theoretically learnt. In these exercises, the student will face real interaction with virtual systems, devices and environments in order to get trained on predefined tasks. Particularly in the SLIM-VRT project, the case studies deal with specific actions or protocols typically carried out in a tanker, ranging from simple familiarization with some ship areas (walking through its decks, rooms and corridors), to complex panel based simulations (like for example the implementation of a secure protocol for loading a particular tank at the port).

The rest of this section presents the general structure of the case studies, and the different VRT functionalities integrated in the context of one of them as an example of the actual implementations in the SLIM-VRT project.

6.1. Case Study General Structure

The learning process in a case study is organized hierarchically following an increasing level of complexity in the operations the student has to carry out inside the virtual environment. Typically it consists of four steps, each one integrating different VRT functionalities depending on the

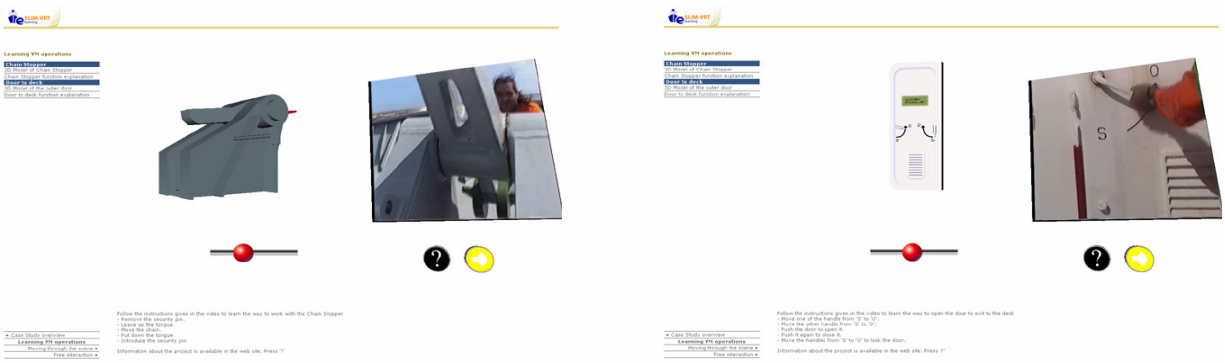

Figure 4: *Manipulating devices: the chain stopper (left) and the door to the main deck (right)*

learning requirements and educational material introduced by the case study designer:

Case Study Overview. The objectives, detailed description and expected results of the case study are presented to the student in textual form.

Devices Operation. In all case studies, the student has to interact with complex devices or elements which may require some specific skills to be able to be operated. In this phase, the student has access to independent 3D models of the relevant devices for the case study (Figure 4), to learn how to interact with them. In addition, he has access to multimedia information integrated with the virtual models about their technical characteristics. In case those devices are to be found integrated in different consoles or panels in the real scene, they are also presented in that way to the student, together with any educational multimedia information relevant for the learning process.

Guided Tours. Moving within a complex virtual scene requires some training in which the student needs to understand the paths to be followed for reaching the different elements involved in the case study. To teach how to carry out those operations, one or more predefined tours may be implemented: the user is moved automatically through a set of different points (stops) where he must interact with the devices introduced in the previous phase of the case study. The predefined tours enhanced functionality is integrated here together with the presentation of information at each stop so that the student can inspect and learn the best procedure to move and act according to the protocols learnt in the course the case study is associated to.

Practical Exercise. The student is finally asked to perform the exercise (main target of the case study) and reproduce effectively the actions learnt through the guided tours. He must interact with the required devices in real time, taking into account their state of operation and the final objectives of the action to be carried out. The student's actions are monitored on-line at the server side to check, validate and correct (if need be) the progress of the exercise. Location and Time events are continuously sent to the server through the IM subsystem to identify the user position and evolution. Particular actions (interactions with devices or elements) are also sent to the server which processes them according to the behaviour logic implemented in the VRIM. The scene then reacts to the user's intervention through that logic, which simulates the real operation of the different devices in the context of the case study.

6.2. Implementation Example

We briefly describe here a particular case study, the chain stopper case study, which integrates the different functionalities provided by the VRT, following the overall structure presented above.

Case Study Overview. The main objective of this case study is to teach the procedure to operate a mooring device called chain stopper and placed on the main deck of the tanker, following specific instructions about the number of links that have to be released depending on the initial conditions of the exercise. The student (an officer), will be placed in one of the most relevant rooms of the ship, the cargo control room, and will be asked to go to the mooring device to carry out a specific operation. There will be time limitations so that the officer will have to learn previously in detail how to go in an efficient manner (but following a *safety path*) from the cargo control room to the front part of the main deck.

Devices Operation. The learning objective of this step is to present the models of the most important devices that the student will find in the practical exercise.

This objective is implemented through 3D models of the main devices and multimedia explanations of the procedures to operate them. The multimedia information provided is integrated with the 3D scenario and it is adapted to the user and terminal profiles through the functionalities provided by the VSD subsystem.

Figure 4 shows two examples of relevant devices whose operation has to be learnt by the officer. In both cases, the graphical model displayed in the middle of the screen can be rotated and the user can interact with it to reproduce the operation indications provided by a video displayed on the right side integrated within the virtual environment. Text explaining in detail the operations is also provided within the web frame below the graphical models.

Guided Tour. The learning objective of this step is to present and explain the safety path the officer should follow to go from the cargo control room to the chain stopper to carry out the exercise.

Figure 5: *Guided tour of the chain stopper case study: from the cargo control room to the chain stopper*

It is implemented using a textual description of the task and enhanced functionalities integrated in a 3D graphics scene, supporting indoors and outdoors simulations. A guided tour drives the student towards the chain stopper, with several stops at relevant points of the path. These stops include multimedia information associated to significant elements or devices, and that information is dynamically adapted to the user/terminal profile.

Figure 5 shows four snapshots taken during the guided tour of the chain stopper case study. From left to right and top to bottom: the initial point, accessing the main corridor from the cargo control room; one stop at the main deck just in front of a shelter, with multimedia information available through an interactivity element (the question mark); when interacting with it, the available information is presented in a menu and an arrow indicates to continue the tour; the final stop is the chain stopper that the student can also interact with.

Practical Exercise. The student, initially placed in the cargo control room, is asked to go to the chain stopper, where he must release two links.

This task is implemented by allowing the student to move freely all along the tanker rooms, corridors and decks. The top left corner of Figure 6 shows a snapshot of the exercise. The avatar's position at any time and his actions within the virtual environment are monitored and evaluated through the functionalities provided by the IM subsystem. All his actions are sent to the server to be processed accordingly (Figure 6-top-right shows the console at the server: events are received and the position, time and actions carried out by the student are registered). The server process could, for instance, modify the illumination conditions (as shown in the bottom row of Figure 6) to increase the complexity of the exercise while the student is moving towards the mooring device.

This shows how the external management of the virtual scene interactivity supported by the VRT allows dynamic modification of the scene conditions in real time together with on-line evaluation and assessment procedures.

7. Conclusions

In this paper, we have described the design and implementation of the Virtual Reality Tool (VRT), a core component inside the educational platform being developed within the EU research project IST-2001-33184 SLIM-VRT (Self Learning Integrated Methodology – Virtual Reality Tool).

The VRT architecture, presented in this paper, contributed to fulfil the user's requirements and learning needs identified within this application field. Different specific developments have been carried out to overcome the limitations that were detected as inherent to the generation of a low-cost application based on VRML, as ours had to be. These developments were designed for providing new enhanced functionalities to the students (such as the generation of predefined tours or the dynamic adaptation of the contents associated to the 3D models according to some session profiles), and for maintaining the efficiency and flexibility of the whole system.

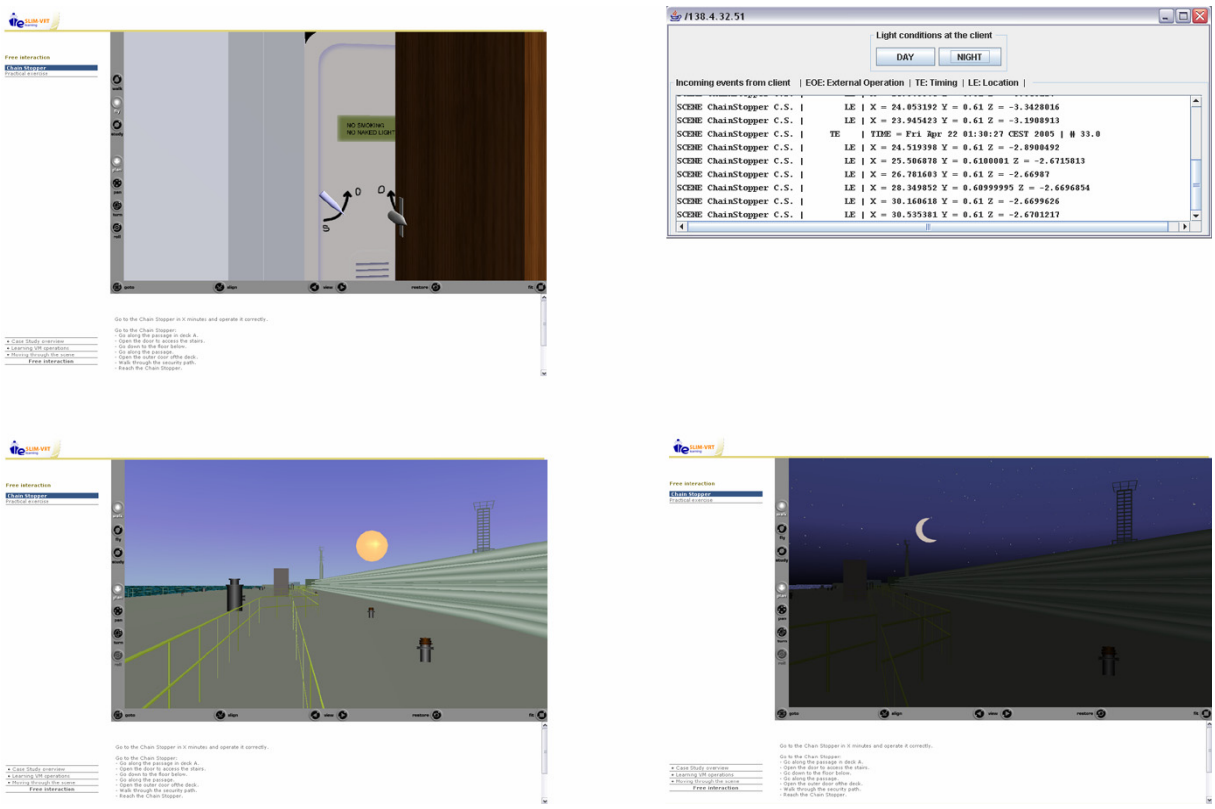

Figure 6: *Practical exercise of the chain stopper case study*

The integration of this VRT within the platform has improved the learning experiences of the students accessing the e-learning application, via inclusion of interactive 3D graphics providing new educational capabilities.

This enhanced content has been organized via different case-studies, which can be basically understood as exercises within topics of specific courses, where the student has to put in practice concepts and protocols theoretically learnt, but now facing real interaction with virtual systems, devices and environments to carry out predefined tasks.

8. Bibliography

[BPT00] Bouras, Ch., Philopoulos, A. and Tsiatsos, Th.: *A Networked Intelligent Distributed Virtual Training Environment: A first Approach.* Int. Workshop on Intelligent Multimedia Computing and Networking (IMMCN 2000), 2000.

[BT04] Bouras, C. and Tsiatsos, Th.: *Distributed VR: building a multi-user layer for the EVE Platform.* Elservier. Journal of Networks and Computer Applications, No. 27, 2004.

[CBR*02] Chover, M., Belmonte, O., Remolar, I., Quirós. R. and Ribelles, J.: *Web-based Virtual Environments for Teaching.* Eurographics/SIGGRAPH Workshop on Computer Graphics Education (CGE 02), 2002.

[HW96] Hartman, J. and Wernecke, J.: *The VRML 2.0 Handbook.* Addison-Wesley, 1996.

[ISO97] ISO/IEC 14772-1:1997. *The Virtual Reality Modeling Language. Part 1: Functional specification and UTF-8 encoding.* ISO/IEC, 1997.

[ISO04] ISO/IEC 19775:2004. *Extensible 3D (X3D).* ISO/IEC, 2004.

[Sli01] SLIM-VRT (IST-2001-33184) EU-funded research project. http://www.gti.ssr.upm.es./~slim/.

[WRO04] Wolff, R., Roberts, D.J. and Otto, O.: *A study of Event Traffic During the Shared Manipulation of Objects within a Collaborative Virtual Environment.* The MIT Press. Presence, **13**-3, 2004.

[WW01] Waller B. and Wilson J.: *A Definition for E-learning.* BAOL Open Learning Today, Issue 58, October 2001.