Adapting Multimedia Information Association in VRML Scenes for E-Learning Applications

Enrique Estalayo, Luis Salgado, Francisco Morán and Julián Cabrera

Grupo de Tratamiento de Imágenes
ETS. Ing. Telecomunicación
Universidad Politécnica de Madrid
28040 Madrid - Spain
+34.913.367.353

{eeg,lsa,fmb,jcq}@gti.ssr.upm.es

ABSTRACT

This paper presents the design and implementation of an efficient strategy for adapting multimedia information associated to virtual environments in the context of e-learning applications. The VRML representation of a scene includes, along with its geometric description, a full specification of the information associated to the 3D models and the way to present it to the student. Such a mechanism implies that the virtual scene has to be reprogrammed or replicated each time the parameters of the learning session are modified, either by the instructor or to adapt the scene according to different students' profiles. We propose a new strategy which consists in separating the multimedia information to be associated to the virtual scene from the 3D models themselves. This allows to manage externally that information, hence overcoming the identified limitations, while additionally increasing design efficiency and content adaptation. Furthermore, the mechanism proposed is based on readily available and platform-independent web technologies so that it can be exploited in other types of 3D graphics standards, such as X3D.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems - *Artificial, augmented, and virtual realities*; K.3.1 [**Computers and Education**]: Computer Uses in Education - *Distance learning*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems - *Distributed Applications*.

General Terms

Algorithms, Management, Performance, Design, Human Factors, Standardization.

Keywords

Virtual environments, e-learning applications, multimedia information association, adaptability, session profile, client-server architecture, adaptability, VRML, Java, XML.

1. INTRODUCTION

E-learning can be defined as the effective learning process created by combining digitally delivered content with learning support and services [17]. 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, which provides many communication capabilities allowing students to have an ubiquitous access to the contents of e-learning applications. Furthermore, the inherent multimedia characteristics of the most common web browsers, which can interpret and render interactive 3D scenes described in standard languages such as VRML [16][8] and X3D [14] thanks to publicly available plug-ins, enable the combination of different media within an e-learning application to improve its contents and thus the corresponding learning process.

3D scene modeling languages, like VRML, are a very powerful asset for e-learning applications, as students are immersed inside virtual worlds which provide the multimedia information specially designed to improve their learning process. Nevertheless, accessing this multimedia information via Internet can easily be translated, from the student's viewpoint, into having to download huge files before being able to start visualizing them. This problem is not only inherent to 3D graphics applications, but to any other rich and potentially bulky content that cannot be streamed and that might be associated to the different virtual models to enhance the information provided to the student by the virtual scene (e.g. an audio clip could be played when the user clicks on the 3D model of an ancient musical instrument [11]). Besides, not only the bandwidth of the connection to Internet should be considered when developing a virtual scene for an e-learning application, but also other parameters related to the students themselves, such as their language or course level.

Despite their obvious potential [12] to improve the e-learning process, the VRML worlds available on the web have so far been relatively limited in terms of scalability and adaptability. This is the case, for example, of 3D graphics based tourist [13] or e-learning applications [15] where additional multimedia information is associated to a virtual environment and where the user is allowed to move freely to access this information. Changes in the user interests, knowledge level or communication characteristics might 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.

Currently, the functionalities provided by the VRML standard for implementing this updating process imply the need to generate different descriptions of the same virtual world, one for each of the different supported profiles. Nevertheless, a minimum adaptation can be achieved by hardwiring different links, which are targeted at the files containing the multimedia information that is going to be adapted depending on the session parameters. However, each time these parameters change, the VRML code has to be replaced and re-downloaded at the client site to achieve the adaptation of the rendered scene. This implies severe limitations in the application: long waiting intervals, pausing 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 overcome the limitations derived from the need of adapting multimedia content to the session parameters of the users, we propose to decouple the information associated to the models from the VRML description of the world. For this purpose, we have developed a new strategy, avoiding the use of commercial solutions, which consists in developing an external logic that will be in charge of managing the association and adaptation processes. Additionally, this approach will reinforce the reusability of the 3D models (since they do not have to be readapted to different contents) as well the flexibility in the design.

The strategy proposed in this paper is being used in the on-going EU-funded research project "Self Learning Integrated Methodology – Virtual Reality Tool" (SLIM-VRT, IST-2001-33184). The rest of the paper is structured as follows: Section 2 introduces the functional description of the proposed strategy, Section 3 presents how it has been implemented for a particular e-learning application and, finally, conclusions and future work trends are listed in Section 4.

2. RELATED WORK

Although in the past most research concerning 3D worlds has been devoted to solve effectiveness and efficiency problems related to the rendering of static 3D scenes, in the last few years significant research has been focused on accessibility and usability issues for interactive 3D environments [2]. Nevertheless, adaptation and personalization of 3D worlds is a less explored issue, even though there are several remarkable studies.

There is an extensive research about how the personalization of 3D content can improve the results pursued in different applications (e.g., the educative process in e-learning platforms [1]). In this sense, first works were devoted in such different fields as Augmented Reality in virtual museums [14] or e-commerce applications [4]. However, one shortcoming of these proposals is that the adapted contents are limited to textual labels or to the models themselves.

Other works, like the one described in [3], present strategies for adapting the interaction level anticipating the user behaviors by studying his/her interaction patterns. Nevertheless, although it is interesting to adapt the interactivity regarding parameters such as hardware capabilities, this strategy is not completely applicable within e-learning applications, as the proper student interactions are one of the tasks to be evaluated.

Recent works propose adaptive architectures for generating dynamically personalized 3D worlds, such as [4] and [5]. This way, proposals like [4] discuss examples of 3D virtual stores, whose content is generated according to the monitorization of the client activity.

We propose to deepen into this last approach by reaching the adaptation not only for the interaction [3] and VMRL models [4], but also for the information associated to them and how it is presented to the user. Furthermore, as this strategy will be integrated within an e-learning platform, the customization of the contents will be gathered according to different student profiles, which will be suitable for being updated within the established sessions.

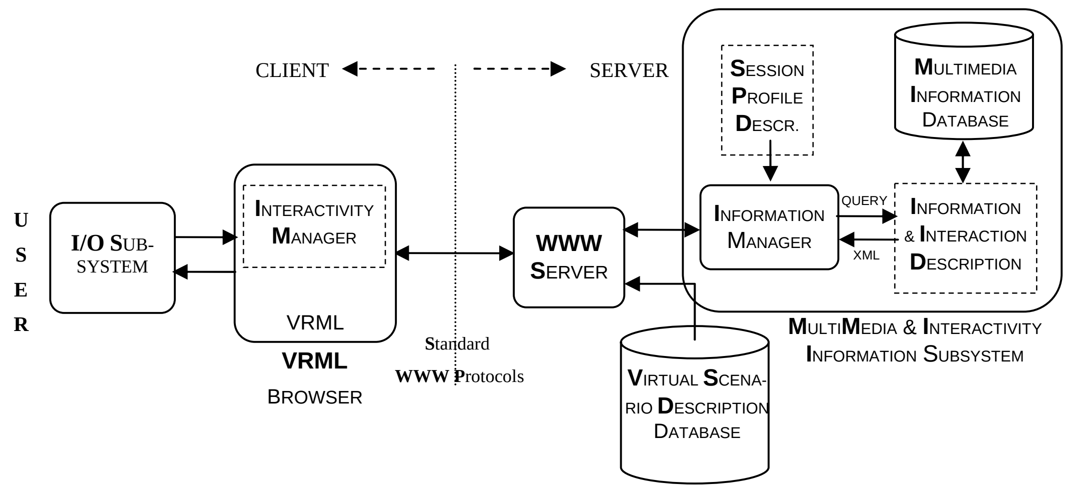


Figure 1. Our client-server system architecture for adaptive VRML applications, whose new elements with respect to the standard one are the *Information manager* and the *Information and Interaction description* in the server, and the *Interactivity manager* in the client.

3. SYSTEM ARCHITECTURE OVERVIEW

The VRML standard [16] offers simple and limited mechanisms to control the interactivity and to associate multimedia information to the modeled virtual worlds. The information can be hardwired inside the geometric description of the scene, and the user will have access to it through a standard navigation interface. Nevertheless, such a simple requirement as to support multilingualism within the application, will require most of the times to have inflexible virtual worlds, where the links have been specified in the code to access the file containing the information in the appropriate language.

To overcome these limitations, new mechanisms are required to dynamically customize the associated information to some external parameters. In our proposal, these mechanisms are implemented introducing new modules within some of the basic subsystems that conform a generic client-server system architecture for e-learning applications with 3D graphics interaction through VRML. The architecture we propose is illustrated in Figure 1.

The server is composed by three subsystems:

1. The *Virtual Scenario Definition database (VSDd)* stores the VRML data suitable for being downloaded to the client during the 3D graphics based e-learning session. Until now, the current client-server applications involving any VRML contents had to transmit the files stored in this *VSDd* each time a session parameter was modified (e.g., with a new session initialization); to overcome this limitation, the *VSDd* incorporates, together with the graphical models of the different elements composing the virtual scene (geometry, textures, etc.), other VRML models in charge of providing further adaptability of the associated information to the 3D graphics application. These VRML models will be identified in this paper as *Interactivity Elements*.
2. The *WWW Server subsystem (WWWSs)* handles the standard communications with the different clients. The graphical models of the virtual scenario are downloaded to each client once the session is initialized.
3. The *Multimedia & Interactivity Information subsystem (MMIIs)* contains the files storing the information to be associated to the VRML models. In our strategy, the *MMIIs* also stores the different modules in charge of achieving the independency and adaptability between the information and the VRML models of the *VSDd*.

At the client side, two different subsystems are identified:

1. The *Browser* is the client application that interprets the downloaded scene description and renders it. Our *Interactivity Elements* are downloaded within the VRML files and rendered too by the *Browser*, handling the communication with the *MMIIs* to control the session parameters and providing information about the actions executed by the user inside the 3D scene.
2. The *I/O subsystem (IOs)* handles the user's interactions with the application at different levels. It can include different devices for the visualization of the virtual world (e.g., a screen, 3D glasses, etc.), as well as the elements allowing any kind of user-computer interaction (e.g., mouse, keyboard, etc.). The possibility to have multiple users accessing the application implies that this module implementation will be highly dependent on the available hardware at the client.

Contributions carried out for the developed application are focused on the *MMIIs* (at the server) and the rendered world in the *Browser* (at the client). The functional elements which compose these two subsystems are also illustrated in Figure 1, and a detailed description of them is provided below.

3.1 Multimedia & Interactivity Information subsystem

The *MMIIs* has four different elements:

1. The *MultiMedia Information database (MMId)* contains the information suitable for being associated to the virtual environment. This information has a well-structured organization so that the access and adaptation of the stored data is achieved efficiently.
2. The *Information and Interaction description (IId)* provides the meta-information (i.e., the description) of the different files containing the multimedia data suitable for being associated to the virtual scene. The level of the description depends on the type of application for which the virtual environment has been created. Nevertheless, for the proposed strategy, this description has to gather at least the information needed to manage the associating process (for example, the location of the different data files), as well as which multimedia data corresponds to the different session profiles. As is the case for the *MMId*, the *IId* is also well organized so that any modifications in the module can be implemented efficiently.
3. The *Session Profile description (SPd)* contains the information stored in the platform hosting the e-learning application relative to the different clients and communication channels. Each established session has its own parameters which cannot be ignored as they specify important characteristics for the correct execution of the application. They include:
   - User profile: information that identifies the user (student) accessing the e-learning platform. There are two parameters that have to be considered in the implementation, as they are particularly relevant for this type of application: the language, in which all the information has to be presented to the user (as it clearly affects the interaction with the *MMId*), and the user's level, understood as her/his grade, knowledge or experience (as it will influence the quality and quantity of the information supplied to her/him during the navigation through the virtual world).
   - Terminal profile: information identifying the hardware capabilities available at client site for the visualization of the virtual scene. This information is particularly relevant as it does not only influence the quality of the rendered models, but also the type of multimedia information that can be presented to the user in the current session (e.g., limited hardware cannot render a 3D virtual scene and display a high-resolution video at the same time).
4. The *Information manager* provides the integration of the previous elements and supports the enhanced functionalities proposed in the present paper. It has two main functionalities:

1. To declare which multimedia information is available for the current student's session, according to the values stored in the *SPd* and the *IId*.
2. To establish a bi-directional communication with the student's browser as (s)he tries to access any of the multimedia information associated to the models in the scene.

The Information manager is also the element in charge of the adaptive process. It must detect any variation that may occur either in the parameters of the *SPd*, or in the *MMId*, and will execute the appropriate mechanisms to upload the *IId*. This process should be transparent to the student, who should not notice these mechanisms while navigating through the scene, as no new models have to be downloaded and rendered into the *Browser*.

3.2 Interactivity manager

The *Interactivity manager* is integrated inside the VRML models and, therefore, it is downloaded into the *Browser*. Its main function is to provide new interaction capabilities inside the virtual scene (i.e., new mechanisms have to be implemented for controlling and offering the access to the multimedia information associated to the 3D models, as now the adaptation is achieved in a dynamic manner), as well as to communicate to the *Information manager* any information requested at any time by the user.

This *Interactivity manager* is composed of different *Interactivity Elements*. These elements are implemented as new VRML models and included in the description of the virtual environment, but they are not initially rendered, until there is some information associated to any of the models presented in the virtual scene and accessible to the student. When this occurs, these elements do show the available information (in the form of a menu with different options, icons, etc.), and then the user has the possibility to request the download and presentation of a specific information by interacting with them. Therefore, the *Interactivity Elements* can be understood as enhanced user interfaces for accessing the associated information in a standardized way. Particularly relevant is the fact that, although all the available information is offered to the student, none of it is downloaded to the client but after a specific student's request.

4. SYSTEM IMPLEMENTATION

The proposed architecture and adaptability strategy have been implemented in the context of a 3D graphics based e-learning application for maritime education [15]. In particular, the selected scenario, modeled for this application in VRML, is a real tanker suitable to offer a wide range of educational possibilities for different maritime learning profiles. The users will be able to access a virtual reconstruction of the ship to follow a pre-defined learning experience. During the lessons, the students navigate within selected parts of the virtual reconstruction of the ship, accessing additional multimedia information associated to relevant areas and specific objects and devices, which helps them learn their functionality and interaction capabilities. The type and complexity of the provided information needs to be adapted to different session profiles, as the e-learning platform supports different types of clients. For instance, multilingualism and adaptability to different knowledge levels have to be efficiently handled. Additionally, the multi-client approach requires that the multimedia information and the level of interactivity of the models be adapted as well to different terminal hardware capabilities, as the students are likely to have different visualization devices, graphic acceleration capabilities or communication channels. Besides, some of those parameters may even vary while the student is experiencing the virtual environment, and the implications on the associated information and the interactivity level can be automatically adapted by following our approach.

Details on how the different system modules have been implemented are introduced in the following subsections, focusing mainly on those directly related to the new association mechanisms we propose.

4.1 Virtual Scenario Description and Multimedia Information Databases

A VRML-based hierarchical modeling approach has been used to generate the virtual scenario [13] following the spatial distribution of the real scenario. For the particular case of the tanker, the root of the hierarchy is a VRML file modeling the external part of the ship, composed by the hull, the outside deck and the different façades of the buildings. The next level is composed by the VRML models of the different decks in the ship, accessed only when the user gets inside the buildings. A third level is composed by the VRML models of the rooms associated to each of the decks. Finally, associated to each room, there are VRML models of the furniture, consoles, and other specific devices.

The *MMId* holds the files containing the information, edited by the course creators, suitable for being associated to the 3D models depending on the different session parameters. This database is organized taking into account the type of the information data (images, video, audio, or text), the different formats (HTML, JPEG, GIF, MPEG, etc.), the language and the resolution level. Due to this classification, the association of the multimedia information within the 3D models is more efficient and facilitates the adaptation process to any modification occurring during the established sessions.

4.2 Information and Interaction description

As the insertion of the information associated to the VRML models is now externally controlled, the *IId* has been incorporated in the system. Its main function is to identify what information files and what interaction capabilities are available for each VRML model in the virtual environment, depending on the session parameters.

For each 3D model with multimedia information and/or different interactivity levels depending on the session parameters, a description file implemented in XML [19] is generated. The structure implemented in this e-learning application is composed of two different types of XML files:

- A *model descriptor* file: there is one for each of the 3D adaptable models. Their main functionality is to store the references to where the real files containing the multimedia data are located in the *MMId*, and the values specifying which levels of interactivity are allowed. These references and values are ordered following a predefined structure (e.g., depending on the media type: audio, video, text, etc.) and they will be accessed in the filtering process that will adapt the contents of the application to the session parameters.
- A *model's controller* file: there is one for each of the VRML models with extended information. Its functionality is to spec-

```
<?xml version="1.0" encoding="UTF-8"?>
<SafetyConsole.xml>
 <!--Declares the information enabled for this device-->
 <Audio info="ON">
    <ES active="ON">..\..\info\audio\testaudio.wav</ES>
    <EN active="OFF">..\..\info\audio\demo.mp3</EN>
    <GR active="OFF">..\..\info\audio\safetyconsole.wav</GR>
 </Audio>
 <HTML info="ON">
    <ES active="ON">http://www.gti.ssr.upm.es</ES>
    <EN active="OFF">http://www.bmt.org/research.asp</EN>
    <GR active="OFF">http://www.slimvrt.gr/</GR>
 </HTML>
 <Text info="OFF">
    <ES active="ON">Consola de Seguridad del barco</ES>
    <EN active="OFF">Safety Console of the Main Bridge</EN>
    <GR active="OFF" />
 </Text>
 <Texture info="OFF">
    <ES active="ON">..\..\info\images\Safety_console.jpg</ES>
    <EN active="OFF">..\..\info\images\Safety_console.jpg</EN>
    <GR active="OFF">..\..\info\images\Safety_console.jpg</GR>
 </Texture>
 <Video info="ON">
    <ES active="ON">..\..\info\video\safetyconsole1.mpg</ES>
    <EN active="OFF">..\..\info\video\safetyconsole2.mpg</EN>
    <GR active="OFF">..\..\info\video\safetyconsole3.mpg</GR>
 </Video>
</SafetyConsole.xml>
```

Figure 2. Example of the XML code included in a model's controller file. profile.

ify if the information related to that model is available or not and which level of interactivity is active. This decision will be controlled externally by the *Information manager*, taking into account the session parameters. These files are the result of the filtering process that takes place each time the student requests information associated to a VRML model.

Figure 2 presents an example of a *model's controller* file. The code shows how the information has been structured taking into account the different formats and which ones have been enabled (this is, the ones with the *info* tag set to ON) depending on the session profile. In the next hierarchical level, the information is organized according to the different languages supported (ES for Spanish, EN for English, and GR for Greek) and just the one agreeing with the profile is enabled (this is, tag *active* set to ON). This way, the code in Figure 2 shows that the current session corresponds to a profile in which the hardware capabilities are high (i.e. they can support video streaming), and the selected language is Spanish.

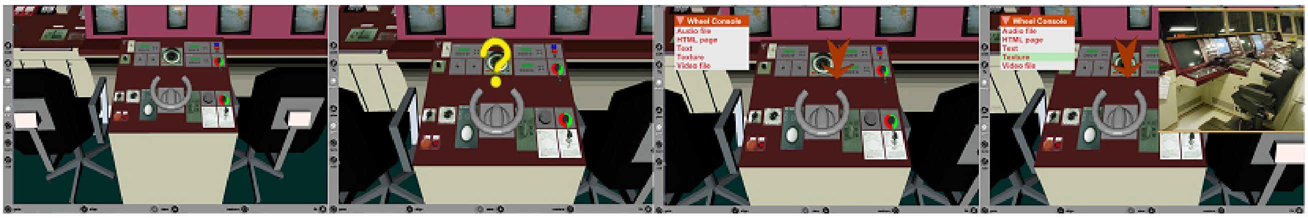
4.3 Interactivity Manager

The *Interactivity Manager* is composed by the already introduced *Interactivity Elements*. They are VRML models that are presented to the user when (s)he approaches any object or element with associated multimedia information. Their content is dynamically adapted to the session parameters, and they handle the communication between the *Information manager* and the client VRML browser using standard VRML communication mechanisms.

The way in which this communication is implemented in the *Interactivity Manager* is using the PROTO node. A PROTO is a VRML functionality that is used in this system to contain a set of mandatory fields, which are accessed both by the client and server, and used to implement an efficient interface between them [12]. This interface allows the model to be rendered as many times as needed, but customizing the parameters specified in the header of the VRML node (e.g., the references to the files containing multimedia data will vary and can be determined by the server) from one rendering to another.

The *Interactivity manager* controls the access to the multimedia data related to the 3D models, so that it should be activated only when the navigation through the virtual world results into a situation in which these data could be requested. Although the proposed strategy supports different approaches (e.g. direct interaction with the models), in the current implementation a three steps interactivity model has been followed for accessing the information associated to the VRML elements:

1. In the first step, the activation of the *Interactivity Elements* depends on the position and orientation of the avatar inside the 3D scene with respect to the position of the models having any associated information. When the user gets close enough to one of these relevant objects, an *Interactivity Element* is displayed (e.g. a question mark, as in Figure 3(b)) to indicate that information associated to the object is available.
2. In the second step, if the user interacts with this new rendered element, an event is sent to the server requesting details on the available information associated to the object. These events are processed by the server and the resulting information, adapted to the current session parameters, is sent back to the client, where a new *Interactivity Element* is displayed (e.g., a menu, cf. Figure 3(c)) containing details on the available information, which has been already adapted.
3. Finally, if the user selects one type of information to be displayed, this information is downloaded from the server, rendered in the *Browser* and presented through the Input/Output subsystem (Figure 3(d)).



(a) **(b)** **(c)** **(d)**

Figure 3. Example of the process followed by the student for accessing the information associated to the models by the Interactivity Elements

4.4 Information Manager

The adaptation of the associated information and interactivity levels is carried out by the *Information manager*. Nevertheless, this module is also in charge of processing the different events generated by the clients during the established sessions.

The *Information manager* has been implemented using Java and, as its name suggests, its main functionality is to manage the process of adapting and associating information to the virtual environment. With this purpose, this module carries out two different processes:

1. **Filtering:** this process adapts the contents stored in the *IDm* so they can be used by the VRML models. It runs each time a client requests any multimedia information associated to any of the 3D models integrated in the virtual environment; this is, each time the student clicks on the *Interactivity Element* representing a question mark. Three different steps are followed by the *Information manager* while executing this process:

    1. The first step is to obtain the parameters stored at the *SPd*, which specify the interactivity level and the types of information available for the current session. The implemented way of communicating these parameters to the *Information manager* is through a Java Servlet [7].

    2. The next step is to access the *IId*. For this purpose, the Java program calls a SAX parser, installed previously within the JDOM classes [1] for establishing the communication between Java and XML, which runs through the *model descriptor* file related to the specific 3D model and extracts the information values from the XML code.

    3. The third step consists on generating the *model's controller* XML file, according to the values and with the parameters obtained from the *SPd*. This XML file is the one containing the information that will be communicated to the VRML scene to control the interactivity and the associated information of the 3D model and, therefore, it will be downloaded into the client.

2. **Adaptation of the contents associated to the virtual world:** during the visualization of the VRML scene, different modifications involving the association of information can take place whenever a session parameter is modified (e.g., the student is upgraded to another level) or whenever something changes inside the *IId* (e.g., the supervisor of a course enables a new multimedia information available for a 3D model). In these situations, the system has to react in real time to readapt the associated information and interactivity levels.

During the implementation of the proposed strategy, different approaches were considered (e.g., to use a periodical thread [6] with the only purpose of checking if any session parameter had changed), but the final choice was to develop a four steps implementation, as it was a more efficient and less complex solution. The developed process begins each time the next *Interactivity Element*, the textual menu, is rendered.

1. First, the *Information manager* receives from the *Browser* an information request, which is encapsulated as a VRML event. For this purpose, Java enables different standardized libraries (vrml.*, vrml.field.* and vrml.node.* [9]).

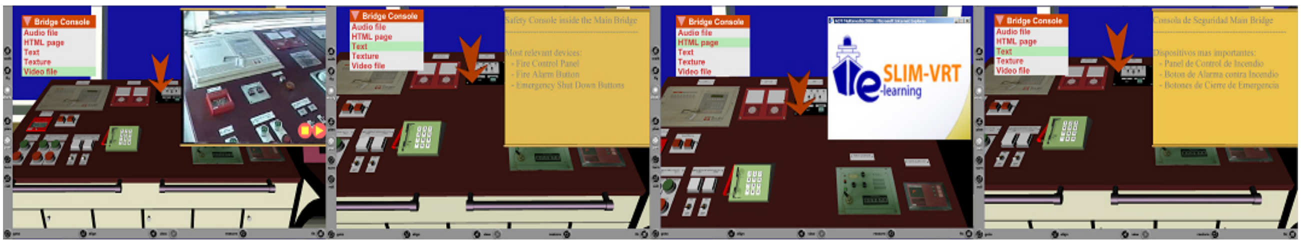
2. Next, the Java class executes the SAX parser to access the parameters stored in the *model controller* XML file as the result of the filtering process.

3. In the third step, the *Information manager* transforms these values into VRML events and sends them back to the *Information Element* (i.e., the textual menu) of the client that has requested the information associated to the model. This way, this *Information Element* can enable the access to the multimedia information, available for the established session and for that concrete 3D model.

4. Finally, the *Interactivity Element* is rendered into the scene and completely customized according to the session parameters and to the information available for that specific 3D model. When the student clicks on any of the options offered in the menu, the information associated to that link is downloaded from the server and rendered in the *Browser*.

    Figure 4 presents some examples of this final step and how the multimedia information is displayed inside the virtual scene. On one hand, Figure 4(a) shows the layout of a video and how it is displayed together with the stop and play buttons If Text is the option selected in the menu, the layout would be similar to the one presented in Figures 4(b) and 4(d). Finally, when the requested information is HTML, a new browser's window is opened, as shown in Figure 4(c).

There is only one more important issue that has to be highlighted at this point, and it is the fact that from the user's point of view, the filtering and adaptation processes are completely transparent as there is no need to reload the 3D graphics into the *Browser* each time the session parameters are modified. The student continues navigating through the virtual scene and, whenever (s)he tries to access any additional multimedia information, the adaptation is achieved as both processes are executed in parallel with the visualization of the virtual world.



(a) **(b)** **(c)** **(d)**

Figure 4. Screenshots of the browser interface after the information associated to the same 3D model is displayed. For this 3D console, the associated information can be a video (a), plain text in different languages (b and d), or an HTML web page (c).

5. EVALUATION

As stated, the design efficiency and content adaptation for the different student profiles are relevant issues regarding e-learning platforms. In our system, different tests have been carried out in which the adaptation to different session parameters and the computational load have been evaluated.

The scenario considered for the testing was composed by an HTML web page, in which the different session parameters could be specified, and a model representing one of the consoles in the main bridge of the tanker, which had different types of information associated.

During the tests, an increasing number of clients (from 1 to 8) with random profiles accessed the server at the same time. The results proved that the adaptation was achieved in real time. This issue was particularly remarkable regarding the language and hardware capabilities parameters, showing how the dynamic association of information had been achieved improving the learning process for the students as it had been adapted to their particular needs. On the other hand, no overload problems appeared in the server during the different sessions.

Finally, the modular architecture improves the system efficiency, allowing designers to re-elaborate the different subsystems and integrating them without further limitations.

In conclusion, the system implementation has demonstrated that the strategy proposed is applicable within an e-learning platform, without inserting conflicts with the different profiles or computational overloads within the server.

6. CONCLUSIONS

In this paper, we have both proposed a general architecture for the adaptation of the multimedia contents related to an e-learning platform and shown its application in the "Self Learning Integrated Methodology – Virtual Reality Tool" (SLIM-VRT, IST-2001-33184) EU-funded, still undergoing, research project.

The architecture presented here helps to flexibly and efficiently adapt the multimedia information associated to the virtual scenes. Modifications in the contents or in the student's session parameters do not imply modifications in the VRML geometric description, as the adaptation process is externally managed. This enables transparency for the student and allows the parallel development of the model geometry and the contents generated within the e-learning courses.

Different tests have been performed showing the correct functioning of the developed modules inside the e-learning platform. Nevertheless, we are currently proceeding with more evaluation tests, specially focused on the final users, the students. Also, more future work is foreseen in the following directions:

- Automatic adaptation of the student's profile: currently the profile parameters are just received from the e-learning platform, but it could be probably very interesting if the student's actions could be evaluated and an automatic feed-back with the server updated the stored profile.
- Porting to X3D: decoupling the multimedia information from the geometric description of the models, as well as the external management of the adaptation process, imply that a migration of the system to support scenes described in X3D should just require a modification of the interfaces.

7. ACKNOWLEDGEMENTS

This work has been partly supported by the Plan Nacional de Ciencia y Tecnología of the Spanish Government under project TIC2001-3069.

8. REFERENCES

[1] Brusilovsky P.: *Adaptive and Intelligent Technologies for Web-based Education*, Küntzkiche Intellligenz, Special Issue on Intelligent Systems and Teleteaching, 1999.

[2] Brutzman, D.: *Teaching 3D Modeling and Simulation: Virtual Kelp Forest Case Study*, Proceedings of the 7th International Conference on 3D Web Technology, 2002.

[3] Celentano, A., Nodari M., Pittarello F.: *Adaptive Interaction in Web3D Virtual Worlds*, Proceedings of the 9th Conference on Web3D Technology, 2004.

[4] Chittaro L. and Ranon R., *Dynamic Generation of Personalized VRML Content: a General Approach and its Aplication to 3D E-Commerce*, 7th International Conference on 3D Web Technology, ACM Press, New York, 2002.

[5] Chittaro L. and Ranon R., *Using the X3D Language for Adaptive Manipulation of 3D Web Content*, Proceedings of A-H: 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 2004.

[6] Eckel B., President MindView Inc: *Thinking in Java*, Ed. Prentice Hall, 2000.

[7] Hall M. and Brown L., *Core Servlets and JavaServer Pages, 2nd Edition*, Ed Sun Microsystems Press/Prentice Hall.

[8] Hartman J. and Wernecke J., *The VRML 2.0 handbook : Building moving worlds on the web* , Addison-Wesley, 1996.

[9] Lea R., *Java for 3D and VRML Worlds*, New Riders, 1996.

[10] McLaughlin, B.: *Java and XML, 2nd Edition*, O'Reilly, 2001.

[11] ORION (IST-2001-39122) EU-funded research project. http://www.gti.ssr.upm.es./~orion/, 2004.

[12] Rischbeck T. and Watson P., *A Scalable, Multi-user VRML server*, Proc. of the Virtual Reality Conference, USA, 2002.

[13] Salgado L., Rendón E., and Artola R., *System Architecture for Touristic Orientation: The TOSCA High-End System*, Proc. of the Int. Conf. Virtual Reality, Archeology, and Cultural Heritage, Glyfada, Greece, 2001.

[14] Sinclair P., Martinez K.: *Adaptive Hypermedia in Augmented Reality*, 12th ACM Conference on Hypertext and Hypermedia, 2001.

[15] SLIM-VRT (IST-2001-33184) EU-funded research project. http://www.gti.ssr.upm.es/~slim/, 2004.

[16] ISO/IEC 14772-1: 1997 Virtual Reality Modelling Language (VRML97), 1997.

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

[18] ISO/IEC 19775:2004. *Information technology –Computer graphics and image processing – Extensible 3D(X3D)*, 2004.

[19] XML 1.0 Specification. http://www.w3.org/TR/2000/REC-xml-20001006, 2004.