ICT for EU-India Cross-Cultural Dissemination
Co-financed by the European Commission

PROCEEDINGS

LET-WEB3D 2004

First International Workshop on Web3D Technologies in Learning, Education and Training

Udine, Italy
September 30 – October 1, 2004

http://hcilab.uniud.it

University of Udine
Dept. of Math and
Computer Science

In cooperation with:

First International Workshop on Web3D Technologies in Learning, Education and Training

PROCEEDINGS

General Chair

Roberto Ranon, HCI Lab, University of Udine (Italy)

Program Committee Chair

Luca Chittaro, HCI Lab, University of Udine (Italy)

Program Committee

Liliana Ardissono, University of Torino (Italy)

Christian Bouville, France Telecom R&D (France)

Paul Brna, University of Northumbria (UK)

Noëlle Carbonell, Université Henri Poincaré (France)

Raimund Dachselt, Technical University of Dresden (Germany)

Fiorella de Rosis, University of Bari (Italy)

Holger Grahn, Fraunhofer Institute (Germany)

Nigel John, University of Wales (UK)

Jan Kautz, Massachusetts Institute of Technology (USA)

Antonio Krüger, Saarland University (Germany)

Nadia Magnenat-Thalmann, MIRALab (Switzerland)

Igor Pandzic, University of Zagreb (Croatia)

Fabio Paternò, C.N.R. – ISTI (Italy)

Mark Pullen, George Mason University (USA)

Ed Sims, Vcom3D, Inc. (USA)

Local Organizing Committee

Stefano Burigat	
Demis Corvaglia	
Luca De Marco	HCI Lab, University of Udine (Italy)
Lucio Ieronutti	
Milena Serra	
Augusto Senerchia	

Foreword

These proceedings contain the papers presented at the *First International Workshop on Web3D Technologies in Learning, Education and Training* (*LET-WEB3D*), which was held September 30 – October 1, 2004 in Udine, Italy.

The purpose of the LET-WEB3D workshop is to bring together researchers and developers from different communities (in particular, Web3D and e-learning), providing a forum to discuss the challenges and potential solutions for effective combination of 3D environments, Web technologies, innovative user interfaces and intelligent tutoring systems.

We received 13 paper submissions from 8 countries. Each paper was sent to three members of the Program Committee for reviewing. Based on reviewers' comments, 9 papers have been accepted.

We would like to thank all the authors, reviewers and local organizing committee members who devoted their time and energy to make this workshop successful.

Roberto Ranon
General Chair

Luca Chittaro
Program Committee Chair

Table of Contents

KEYNOTE SPEECH – Abstract

Web3D Technology in Learning, Education, and Training: The 2-Sigma Challenge

Edward M. Sims, Ph. D.,
Chief Technology Officer, Vcom3D, Inc., USA

Over 20 years ago, researchers discovered that, for a wide domain of instruction, one-on-one tutoring could achieve a 2-sigma improvement in results, as compared to conventional classroom lecture. Simply put, a two-sigma increase in instructional effectiveness means that a 50th-percentile student taught by an individual tutor will learn at the rate of a 98th-percentile student taught in a traditional classroom setting. Today, using the Web, we have the technology to deliver individualized instruction to any student at any time and place. However, the effectiveness of this instruction seldom approaches that of one-on-one tutoring. Web3D technology presents many opportunities to address the "2-sigma challenge" by enhancing not only the cognitive, but also the experiential and affective aspects of learning. In this presentation, we speculate on how Web3D may best combine forces with Intelligent Tutoring Systems and other instructional methodologies to close the 2-sigma gap.

Edward M. Sims has been a leader in the use of computer simulation technologies to provide advanced human interfaces since 1975. Prior to co-founding Vcom3D, Inc., he provided technical leadership for networked visual simulation as Technical Director for GE Aerospace and Lockheed Martin Information Systems Co. As Chief Technology Officer for Vcom3D, Dr. Sims leads the development of "intelligent" 3D virtual humans that communicate through natural language, are animated and rendered in real-time on low-cost personal computers, and that provide sufficient fidelity to support real-time visual communications by signing, gesture, and facial expression. Dr. Sims has been awarded five patents in the areas of visualization and animation. His most recent patent (#6,535,215), entitled "Method for the Animation of Computer-generated Characters", provides a foundation for Vcom3D's "virtual human" technology. Dr. Sims received an B.S. in Mathematics from the College of William and Mary, an M.S. in Electrical Engineering from Rensselaer Polytechnic Institute, and the Ph.D. in Systems Engineering from Rensselaer Polytechnic Institute in 1976. Dr. Sims is a member of the Association for Computing Machinery (ACM), the Web3D Consortium, the Humanoid Animation (H-Anim) Working Group, and the Simulation Interoperability Standards Organization (SISO).

Papers

A Brief Introduction to Web3D Technologies in Education: Motivations, Issues, Opportunities

Luca Chittaro
HCI Lab – Dept. of Math and Computer Science
University of Udine
via delle Scienze 206, 33100 Udine, Italy
ph.: +39 (0432) 558.450
chittaro@dimi.uniud.it

Milena Serra
HCI Lab – Dept. of Math and Computer Science
University of Udine
via delle Scienze 206, 33100 Udine, Italy
ph.: +39 (0432) 558.450
serra@dimi.uniud.it

ABSTRACT
Web3D open standards (such as VRML and X3D) allow the delivery of interactive learning environments through the Internet, reaching potentially large numbers of learners worldwide, at any time. This paper introduces and discusses the educational use of Virtual Reality based on Web3D technologies. We highlight the interesting features of these solutions and summarize the pedagogical basis that motivated their exploitation. We outline the main positive and negative results obtained so far, and point out some of the current research directions.

Categories and Subject Descriptors
K.3.1 [**Computers and Education**]: Computer Uses in Education – *collaborative learning, computer assisted instruction (CAI), distance learning.* I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism – *Virtual reality.*

General Terms
Human Factors.

Keywords
Educational virtual environments, Web3D technologies, virtual learning/training, constructivism.

1. INTRODUCTION
In a continuously changing world, education cannot be restricted to the school years, but has to be a lifelong process. In the last years, the use of virtual reality (VR) as an educational tool has been proposed and discussed by several authors. Virtual environments (VEs) offer the possibility to recreate the real world as it is or to create completely new worlds, providing experiences that can help people in understanding concepts as well as learning to perform specific tasks, where the task can be repeated as often as required and in a safe environment. A considerable problem is that developing and delivering applications with proprietary VR technologies can be very expensive, and thus not accessible to many learners. An emerging solution is provided by Web3D open standards (such as VRML[1] and X3D[2]) that allow the delivery of interactive VEs through the Internet, reaching potentially large numbers of learners worldwide, at any time. Web3D VEs can also be independent from the platform and require only a plug-in for a Web browser to be displayed.

This paper discusses the educational use of Virtual Reality based on Web3D technologies. We highlight the interesting features of these solutions and summarize the pedagogical basis that motivated their exploitation. We outline the main positive and negative results obtained so far, and point out some of the current research directions.

2. EDUCATIONAL CONTEXTS
The flexibility and portability of Web3D technologies allow one to use them in building educational VEs (EVEs) for several contexts:

- *formal education*. This comprises every type of scholar instruction, from kindergarten to college. In this context, EVEs are meant to be used by students supervised by teachers, often during classroom or laboratory lessons;
- *informal education*. This is the context of museums, cultural sites, zoos and similar institutions. The intended users are visitors, possibly helped by a guide;
- *distance or electronic learning*. This comprises both self-instruction through the Web and computer-mediated learning that involves a human teacher interacting with the user through the net;
- *vocational training*. This comprises training in skills required for one's job. Industry, medicine and military are only some of the many domains where training is an

[1] ISO/IEC 14772-1:1997 e ISO/IEC 14772-2:2004 [1]

[2] ISO/IEC FDIS 19775:2004 [2]

every-day practice. Virtual training is meant to be a substitute for on-the-field training at least in the first training phase;

- *special needs education*. People with physical or psychological handicaps require special educational techniques. VEs allow them a wider range of experiences with respect to traditional lessons, even experiences they will not have the chance to try in the real world.

3. PEDAGOGICAL MOTIVATIONS

The basic theory supporting educational uses of EVEs is *constructivism*. Constructivists claim that individuals learn through a direct experience of the world, through a process of knowledge construction that takes place when learners are intellectually engaged in "personally meaningful tasks" [3]. Following this theory, *interaction* with the world is relevant for the learning process. The possibility of providing highly interactive experiences is thus one of the best-valued features of VEs. When we interact with an environment, be this real or virtual, our type of experience is a *first-person* [4] one, that is a direct, non-reflective and, possibly, even unconscious type of experience. On the contrary, *third-person experiences* [4], that are the result of interaction through an intermediate interface (e.g., someone else's description of the world, a symbolic representation, a computer interface that stands between the environment and the user,…), require deliberate reflection and cannot provide the same depth of knowledge as the first-person ones. In many cases, interaction in a VE can be a valuable substitute for a real experience, providing a first-person experience and allowing for a spontaneous knowledge acquisition that requires less cognitive effort than traditional educational practices.

Constructivist theory defines learning as "a social activity that is enhanced by shared inquiry" [4]. Each individual creates her own interpretation of knowledge-building experiences, but it is necessary to develop a shared meaning to allow reliable communication among people. *Collaborative learning* is a solution, and indeed groupwork activity improves personal cognitive development together with social and management skills. Collaborative learning is a learner-centered approach: each learner is expected to participate actively in discussions, decisions, common understanding and goals achievement [5]. Web3D technologies can provide new tools and scenarios for collaborative learning, connecting people that are even physically located in distant places. An example of *collaborative virtual environment* (CVE) is CVE-VM [6], that is aimed to support teaching and learning in Brazilian schools. Children collaborate over the Internet to create their virtual world, thus actively constructing knowledge on the subjects of the world. Following the taxonomy of EVEs proposed by Youngblut [7], CVE-VM is not a *pre-developed application*, where students can only interact with the VE, but it is a *multi-user distributed application*, where students work together not only to solve problems but also to extend the virtual world. The goal of this type of applications is indeed construction and deepening of knowledge, with a focus on collaboration among students.

Traditional educational methods rely on knowledge, acquired by books and teachers, that must be then applied to real situations. The *situated learning* approach, on the contrary, suggests that it is easier for students to learn concepts in the same context where these will be applied. Indeed, "a situated learning environment provides an authentic context that reflects the way knowledge will be used in real-life" [8]. VEs can provide a good level of realism and interactivity, thus they can substitute for real-life and provide situated learning experiences.

Research in human learning processes demonstrates that individuals acquire more information if more senses are involved in the acquisition process [9], i.e. we are more receptive when we see, listen, hear and act at the same time. In VEs, one can exploit this human capability by providing *multisensory stimuli*, such as three dimensional spatialized sound or haptic stimuli (e.g., vibration, force).

4. ADVANTAGES OF WEB3D TECHNOLOGIES IN EDUCATION

Educational uses of Web3D technologies present a number of advantages with respect to traditional learning practices. Firstly, the use of *three dimensional graphics* allows for more realistic and detailed representations of topics, offering more viewpoints and more inspection possibilities compared to 2D representations. With an EVE, one can provide a wide range of experiences, some of which are impossible to try in the real world because of distance, cost, danger or impracticability. For example, it is possible to reconstruct ancient buildings and cities to show how they originally might have looked and how life could be in ancient times, or it is possible to train astronauts on correct procedures before they leave earth. Thanks to Web3D technologies, these EVEs can be made accessible anywhere there is a computer connected to the Internet.

In general, Web3D technologies allow the development of Web-based EVEs that provide the knowledge-building experiences discussed by Winn [4] and related to the concepts of *size*, *transduction* and *reification*. In a VE, users can change their size to gain a better point of view on the explored subject. For example, they can grow until they can see interplanetary spaces or they can shrink until they become able to see atoms and molecules. The concept of *transduction* is somewhat deeper and more complex. A transducer is a device that converts information into forms available to our senses. A VE can convert every type of data into shapes, colors, movements, sounds, or vibrations, i.e. into something that we can see, hear or feel as an haptic sensation. VEs can therefore be considered as transducers that widen the range of information accessible through a first-person experience. Through transduction and changes in size, users can perceive even what in the real world has no physical form. Finally, *reification* refers to the process of creating perceptible representations of abstract concepts. As Winn points out, the above mentioned three kinds of knowledge-building experience "are not available in the real world, but have invaluable potential for education".

Another advantage is the possibility of analyzing the same subject or phenomenon from different point of views. This way, users can gain a deeper understanding of the subject and create more correct and complete mental models to represent it. As an example, Li et al. [10] developed a Web3D training simulator for the treatment of trigeminal neuralgia, a neurosurgical treatment that requires the insertion of a needle under the skin of the patient to puncture the foramen ovale. The interface displays to the trainee two different viewpoints: the external one, which is the surgeon's usual viewpoint, and the internal one, which is the needle's viewpoint. The latter is not realistic, but it helps trainees to localize the foramen ovale and create a correct mental model of its position. A different example of multiple viewpoints is provided by the Virtual Big Beef Creek project [11], where a real estuary has been reconstructed to afford users to navigate, to get data and information in order to learn more about ocean science. Each user can explore the environment using different avatars, each of them allowing different viewpoints and navigation constraints. For example, if a user chooses the scientist avatar, she can move as a human being and acquire data such as water temperature, while if she chooses the fish avatar, she can swim deep in the ocean and can not surface.

Interacting with another human being is definitely easier and can be more appealing than interacting with a book or a computer. As a consequence, the possibility of interacting with virtual humans inside EVEs can be considered as an advantage, whose usefulness in the educational context is manifold. To this purpose, Web3D technologies include the H-Anim[3] standard for modeling three dimensional human figures. First of all, virtual humans can represent in a faithful way the subject of study, as it happens in training simulations for clinicians and first-aid operators. In similar situations, the value added by a virtual patient compared with a dummy is the possibility of physical and, above all, emotional responses, that increase the realism of the scene and the involvement of trainees. Emotions are also important when dealing with virtual teachers, in distance or electronic learning contexts, for example. The mere presence of a lifelike character has proved to have a positive impact on student's perception of the learning experience (the "persona effect" [13]); an even stronger impact on motivation can be obtained by a virtual teacher that shows interest and sensitivity to the student's progress, displaying enthusiasm when the student achieves good results and a sad behavior when the student goes wrong [14]. Virtual teachers, or *animated pedagogical agents*, present other important advantages. First of all, they introduce the social dimension into distance and electronic learning, which are often perceived as cold, impersonal and thus demotivating. Second, they can show how to perform a task instead of simply explaining how to do it and this decreases the educational time, since learning by example is more effective than learning by explanation. Third, pedagogical agents can use nonverbal communication both to enrich explanations and to give feedback to users. For example, they can drive users attention towards an object with a deictic gesture or with gaze, or they can react positively or negatively to users answers through only facial expression. This type of communication is preferable to the verbal one, because it does not interrupt or distract the user [14]. An example of how virtual humans can be used in educational contexts is the Online Mentors for Foreign Language Training and Cultural Familiarization project [15]. During this project, a virtual course to teach elements of Iraqi dialect, nonverbal language, culture and customs has been developed. In this course, virtual humans serve both as *actors*, playing interactive sketches concerning common situations, and as *instructors*, explaining and asking verification questions about language expressions and proper behaviors in the proposed context.

[3] ISO/IEC FCD 19774:2004 [12]

5. LIMITATIONS OF WEB3D TECHNOLOGIES IN EDUCATION

The use of Web3D technologies in educational practices encounters also some problems. Some of them are common to all VEs, such as difficulties in navigation [7] and in the use of interfaces. Users are often not able to move as they want, they get easily lost or do not know how to reach a particular location or point of view. Since educational environments often address non-expert users, both movement abilities and spatial cognition should be made very simple. As far as interface-related problems are concerned, VEs are potentially able to afford a first-person experience as defined by [4]. In fact, several EVEs provide users with a set of possible actions and this is mainly accomplished through an interface, that is not always easy to understand and simple to use.

Other types of problems concern the educational context, such as teachers' lack of experience or difficulties in classroom use. While VEs in vocational training are used to learn and practice how to perform a task before doing it in the real world (and thus teaching methodologies are not very different from traditional ones), the traditional approach in formal education involves a knowledge-bearing teacher that explains concepts to students. EVEs are aimed at fostering active knowledge discovery and construction and the teachers' role changes from education dispenser with all the answers to a companion-guide [7]. Lessons structure also needs to be changed in a consistent way. As a consequence, integrating EVEs in traditional lessons in an effective way is a very difficult task, still under investigation.

Users' disappointment can negatively influence learning: the expectation of the learners can often be too high if they think that the VE will mimic reality, and the lack of realism then detracts from the learning process. It can sometimes be effective to abstract the task to something simpler, that does not aim at faithfully reproducing the real task, but simply at acquiring the skills that are necessary to perform it. An example of task abstraction is provided by the MIST (Minimally Invasive Surgical Trainer) system [16], a surgical simulator for acquiring laparoscopic psychomotor skills. The VE does not faithfully represent organs, which would be a hard task to accomplish, but it abstracts them using approximate geometric shapes.

One particular category of problems is related to the use of immersive VR hardware. An immersive experience can be more effective than a desktop one, but a user placed inside a CAVE or wearing a Head Mounted Display (HMD) is not able to follow written explanations provided by the teacher, nor is she able to take notes or to complete written questionnaires [17]. Moreover, all types of text instructions and information are not suitable for VEs, because 3D is a bad way to display text and HMDs have often low resolutions.

It must be added that Web3D technologies do not presently offer an easy and flexible support to the adoption of immersive hardware and special peripherals. However, Web3D researchers are working at overcoming this limitation. For example, Behr et al. [18] are developing solutions to use VRML/X3D with immersive and augmented reality hardware, while Soares et al. [19] are developing an X3D browser that runs on commodity computer clusters and immersive hardware. An example of commercial product that deals with special pheripherals is the

Reachin Laparoscopic Simulator [20] for the training of surgeons. The Reachin API uses an extended version of VRML 97 where additional nodes have been provided to allow the description of the haptic properties of an object. As a consequence, haptic force feedback is provided to the trainee, improving both the realism and the usefulness of virtual practice. Recently, the Web3D community has promoted the creation of a working group whose aim is the extension and standardization of Web3D technologies in order to deal with different user interface hardware technologies [21].

6. EVALUATING RESULTS

When evaluating an EVE, three main aspects should be taken into account, i.e. understanding, transfer of training and retention. *Understanding* is usually evaluated, and there is a considerable number of positive results reported in the literature. Unfortunately, no standard adequate evaluation method has been developed yet and thus results could not be reliable.

The *transfer of training* from the virtual to the real world can be mainly applied to vocational training or, in general, to sensorymotor tasks. It seems reasonable to think that a simulation can be a valuable substitute for the real world at least in the first period of training. In fact, few systematic empirical studies have been carried out to show this and have not led to clear conclusions about "what sort of training shows transfer, in what conditions, to what extent and how robust the transferred training has proved to be" [22]. This lack of evaluations may be partially due to the lack of cheap, easily delivered EVEs. Web3D technologies could thus give the opportunity to test, also through Internet delivery, more EVEs. A number of Web3D training simulators has been developed for surgery procedures, such as the treatment of abdominal aortic aneurysm [23] or ventricular catheterisation [24]. The evaluation of a simulator for the lumbar puncture, for example, has shown positive results on training of surgeons, who agreed this tool improved their practical skills [25].

Human beings remember positive experiences with more pleasure (and more) than negative ones. Since educational virtual experiences are generally considered more appealing and entertaining than traditional ones, their use should increase the *retention* of acquired knowledge. In fact, there are no long term studies to prove or confute this thesis, because almost all carried out evaluations cover very short time periods of use of EVEs [7].

A very positive result of many evaluations is that users enjoy dealing with EVEs. They are more curious, more interested and have more fun than when learning with traditional methods. As a consequence, they get more involved in their education process and apply more willingly to it (leading some to use the words "edutainment" and "entertainment education").

7. DISCUSSION AND CONCLUSION

During the '90s, several projects have been carried out with the aim of bringing EVEs into educational practice. Despite this effort and reported positive results, EVEs are not yet part of typical educational practices. While in vocational training and informal education contexts we can see a slow but continuous growth of EVEs, the formal education context seems less developed in the last years.

A first reason is related to insufficient funds of educational institutions. In this case, Web3D technologies can play an important role both in reducing costs and in making EVEs more easily accessible to institutes through the Web.

A second issue concerns the justification of the approach. Unfortunately, there is a lack of a sufficient number of validated results: evaluations of EVEs have been generally carried out on small groups of individuals over a short time. Studies involving more users and carried out over long term periods are thus needed to thoroughly assess the benefits and limitations of these solutions. EVEs should be tested as integral parts of curricula, so as to give students and teachers the time to get used to them and to integrate them in everyday practice.

The attitude of teachers towards EVEs and their adoption into classroom activity is another factor. Some teachers may not be interested in new technologies, perceiving them as a waste of time or as a too radical change to their traditional methodology, or simply they may not be familiar with computers and they may not like the fact that their students often have more expertise than them. This issue can be partially tackled by involving teachers in the design of EVEs, by offering them computer training and by developing learning environments that do not require them to demonstrate any expertise.

Another issue concerns proper design of the EVEs, e.g. taking into account both the constructivist theories that have been mentioned in this paper and the usability issues, such as simple navigation abilities.

Finally, the proper integration of EVEs into curricula is an issue in itself. At a minimal level, the 3D experience can deal only with the examples and exercises proposed by a traditional textbook. At a more ambitious level, the 3D environment, from a constructivist point of view, can come before the textbook as the main way to familiarize with the topics. The first approach is more easy to implement and introduce in a classroom, but limits the potential of using EVEs in education. The second approach would require to significantly rethink educational practice. Finding the best trade-off between them is thus an important aspect.

8. ACKNOWLEDGEMENTS

We are grateful to Paul Brna, Nigel John and Ed Sims for the precious comments they provided on a first draft of this paper.

We thank the EU-India Economic Cross Cultural Programme ("ICT for EU-India Cross-Cultural Dissemination" project) for the partial support to the organization of this workshop.

9. REFERENCES

[1] *The Virtual Reality Modeling Language*. 2004. http://www.web3d.org/x3d/specifications/vrml/ISO_IEC_14772-All/index.html (last access on August 2004).

[2] *X3D and Related Specifications*. 2004. http://www.web3d.org/x3d/specifications/index.html (last access on August 2004)

[3] S. Conceição-Runlee, B.J. Daley. *Constructivist Learning Theory to Web-based Course Design: An Instructional Design Approach*. XVII Annual Midwest Research-to-Practice Conference in Adult, Continuing and Community Education, Muncie (Indiana, U.S.A.), 1998.

[4] W. Winn. *A Conceptual Basis for Educational Applications of Virtual Reality*. HITL Report No. R-93-9, 1993.

[5] P. Brna, R. Aspin. *Collaboration in a Virtual World: Support for Conceptual Learning?* In Dicheva, D. and Stanchev, I. (eds.), Proceedings of the IFIP WG 3.3 Working Conference "Human-Computer Interaction and Educational Tools"', pp. 113-123, 1997.

[6] G. Kirner, C. Kirner, A. L. S. Kawamoto, J. Cantão, A. Pinto, R. S. Wazlawick. *Development of a Collaborative Virtual Environment for Educational Applications*. Proceedings of the 6th International Conference on 3D Web Technology, Paderborn (Germany), pp. 61-68, ACM Press: New York, 2001.

[7] C. Youngblut. *Educational Uses of Virtual Reality Technology*. Technical Report No. IDA Document D-2128, Institute for Defense Analyses, 1998.

[8] J. Herrington, R. Oliver. *Critical Characteristics of Situated Learning: Implications for the Instructional Design of Multimedia*. Learning with Technology, ASCILITE'95 Conference Proceedings, J. Pearce & A. Ellis (Eds.), pp. 253-262, 1995.

[9] A. Barraclough, I. Guymer. *Virtual Reality - A Role In Environmental Engineering Education?*. Water Science & Technology, Vol. 38, No. 11, pp. 303-310, 1998.

[10] Y. Li, K. Brodlie, N. Philips. *Web-based VR Training Simulator for Percutaneous Rhizotomy*. Medicine Meets Virtual Reality 2000, J.D. Westwood, H.M. Hoffman, G.T. Mogel, R.A. Robb and D. Stredney (eds.), pp. 175-181, IOS Press: Amsterdam, 2000.

[11] B. Campbell, P. Collins, H. Hadaway, N. Hedley, M. Stoermer. *Web3D in Ocean Science Learning Environments: Virtual Big Beef Creek*. Proceedings of the 7th International Conference on 3D Web Technology, Tempe (Arizona), pp. 85-91, ACM Press: New York, 2002.

[12] *ISO/IEC FCD 19774 — Humanoid animation (H-Anim)*. 2004. http://www.web3d.org/x3d/specifications/ISO-IEC-19774/index.html (last access on August 2004)

[13] J.C. Lester, S.A. Converse, S.E. Kahler, S.T. Barlow, B.A. Stone, R.S. Bhoga. *The Persona Effect: Affective Impact of Animated Pedagogical Agents*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta (Georgia, U.S.A.), pp. 359-366, ACM Press, 1997.

[14] W.L. Johnson, J.W. Rickel, J.C. Lester. *Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments*. International Journal of Artificial Intelligence in Education, Vol. 11, pp. 47-78, 2000.

[15] E.M. Sims, W.Y. Pike. *Reusable, Lifelike Virtual Humans for Mentoring and Role-Playing*. Paper presented at Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), Orlando (Florida, U.S.A.), 2004.

[16] R. McCloy, R. Stone. *Science, medicine, and the future. Virtual reality in surgery*. BMJ, Oct 20, 323(7318), pp. 912-915, 2001.

[17] C. Dede, M.C. Salzman, R. Bowen Loftin, D. Sprague. *Multisensory Immersion as a Modeling Environment for Learning Complex Scientific Concepts*. Computer Modeling and Simulation in Science Education, Nancy Roberts, Wallace Feurzeig, and Beverly Hunter (eds.), pp. 282-319, Springer-Verlag: Berlin, 1998.

[18] J. Behr, P. Dähne, M. Roth. *Utilizing X3D for Immersive Environments*. Proceedings of the 9th International Conference on 3D Web Technology, Monterey (California), pp. 71-78, ACM Press: New York, 2004.

[19] L. P. Soares, M. K. Zuffo. *Jinx: An X3D Browser for VR Immersive Simulation Based on Clusters of Commodity Computers*. Proceedings of the 9th International Conference on 3D Web Technology, Monterey (California), pp. 79-86, ACM Press: New York, 2004.

[20] Reachin. *Reachin Laparoscopic Trainer 2.0*. 2003. http://www.reachin.se/products/reachinlaparoscopictrainer/ (last accessed on August 2004)

[21] *X3D User Interface*. Panel at the 9th International Conference on 3D Web Technology, Monterey (California), 2004.

[22] F.D. Rose, E.A. Attree, B.M. Brooks, D.M. Parslow, P.R. Penn, N. Ambihaipahan. *Transfer of training from virtual to real environments*. Proceedings of the 2nd European Conference on Disability, Virtual Reality and Associated Techniques, Skvde (Sweden), ECDVRAT: UK, 1998.

[23] K. Brodlie, N. El-Khalili, Y. Li. *Key Issues in Developing Web-based Surgical Simulators*. Short paper for the workshop on Web3D for Medical Education and Training at the 8th International Conference on 3D Web Technology, Saint Malo (France), 2003.

[24] N.W. John, M. Riding, N.I. Phillips, et al. *Web-based Surgical Educational Tools*. Medicine Meets Virtual Reality 2001, Studies in Health Technology and Informatics, pp. 212-217, IOS Press: Amsterdam, 2001.

[25] Moorthy, M. Jiwanji, J. Shah, F. Bello, Y. Munz, A. Darzi. *Validation of a Web-Based Training Tool for Lumbar Puncture*. Proceedings of Medicine Meets Virtual Reality MMVR03, IOS Press: Amsterdam, 2003.

Educational Resources and Implementation of a Greek Sign Language Synthesis Architecture

K. Karpouzis
Image, Video and Multimedia Systems Lab
9, Iroon Polytechniou Str., Athens, Greece
+30 210 7723037
kkarpou@cs.ntua.gr

G. Caridakis
Image, Video and Multimedia Systems Lab
9, Iroon Polytechniou Str., Athens, Greece
+30 210 7723038
gcari@image.ntua.gr

S-E. Fotinea
Institute for Language and Speech Processing
Artemidos 6 & Epidavrou, Athens, Greece
+30 210 6875300
evita@ilsp.gr

E. Efthimiou
Institute for Language and Speech Processing
Artemidos 6 & Epidavrou, Athens, Greece
+30 210 6875300
eleni_e@ilsp.gr

ABSTRACT

In this paper we present the creation and presentation of dynamic linguistic resources of Greek Sign Language (GSL). The resources feed the development of an educational multitask platform within the SYNENNOESE project for the teaching of GSL. The platform utilizes standard virtual character (VC) animation technologies for the synthesis of sign sequences/streams, exploiting digital linguistic resources of both lexicon and grammar of GSL. In SYNENNOESE, the input is written Greek text from early elementary school textbooks, which is transformed into GSL and animated on screen. A syntactic parser decodes the structural patterns of written Greek and matches them into equivalent patterns in GSL, which are then signed by a VC. The adopted notation system for the lexical database is HamNoSys (Hamburg Notation System). For the implementation of the virtual signer tool, the definition of the VC follows the h-anim standard and is implemented in a web browser using a standard VRML plug-in.

Categories and Subject Descriptors

J.5 [**Arts and Humanities**]: Language translation, H.5 [**information interfaces and presentation**]: Multimedia Information Systems – *Artificial, augmented, and virtual realities*, H.5 [**information interfaces and presentation**]: Multimedia Information Systems – *Training, help, and documentation*

General Terms

Design, Human Factors, Standardization, Languages.

Keywords

Sign language synthesis, linguistic content, h-anim, educational resources.

1. INTRODUCTION

Greek Sign Language (GSL) is a natural visual language used by the members of the Greek Deaf Community with several thousands of native or non-native signers. Research on the grammar of GSL per se is limited; some work has been done on individual aspects of its syntax, as well as on applied and educational linguistics. It is assumed that GSL as we now know it, is a combination of the older type of Greek sign language dialects with French sign language influence. Comparison of core vocabulary lists exhibit many similarities with sign languages of neighboring countries, while in morphosyntax GSL shares the same cross-linguistic tendencies as many other well analyzed sign languages [1][16].

GSL has developed in a social and linguistic context similar to most other sign languages. It is used widely in the Greek deaf community and the estimation for GSL users is about 40,600 (1986 survey of Gallaudet Univ.). There is also a large number of hearing non-native signers of GSL, mainly students of GSL and families of deaf people. Although the exact number of hearing students of GSL in Greece is unknown, records of the Greek Federation of the Deaf (GFD) show that, in the year 2003 about 300 people were registered for classes of GSL as a second language. The recent increase of mainstreamed deaf students in education, as well as the population of deaf students scattered in other institutions, minor town units for the deaf and private tuition may well double the total number of secondary and potential sign language users. Official settings where GSL is being used include 11 Deaf clubs in Greek urban centers and a total of 14 Deaf primary, secondary and tertiary educational settings.

In consultancy with the Greek Pedagogical Institute, the SYNENNOESE project helps young pupils acquire the proper linguistic background so that they can take full advantage of the new accessible educational material. The platform offers students the possibility of systematic and structured learning of GSL for either self-tutoring or participation to virtual classroom sessions of asynchronous teaching, and its design is compatible with the principles that generally define systems of open and distant learning. Besides teaching GSL as a first language, in its present form the platform can be used for the learning of written Greek through GSL, and it will also be open to future applications in areas of other subjects in the school curriculum.

Figure 1 describes the abstract architecture and dataflow between the components of the integrated system. In this paper we describe the procedures followed during the compilation of the educational material and the implementation of the sign language synthesis component of the educational platform. In this process we utilized existing software components for the web-based animation of an h-anim virtual character; the adoption of widely accepted character definition and animation standards caters for the extensibility and reusability of the system resources and its content.

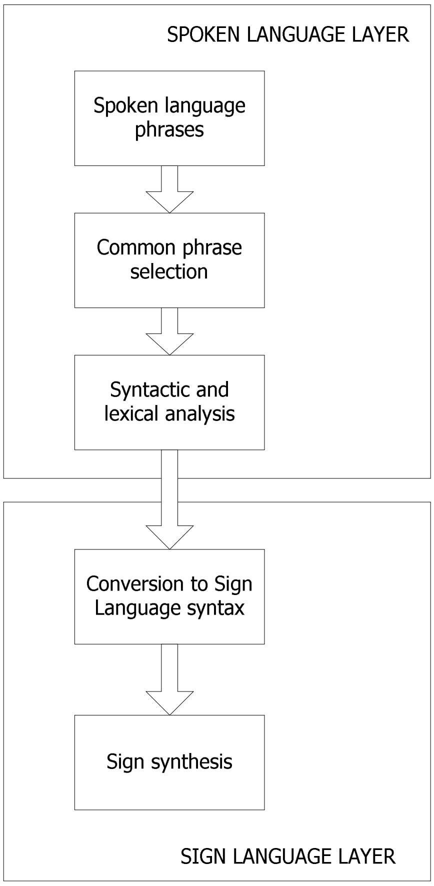


Figure 1: Overview of the proposed architecture

2. LINGUISTIC RESEARCH BACKGROUND IN THE AREA OF SIGN LANGUAGES

In Greece there have been some serious attempts of lexicography in the recent past (PROKLESE, a Dictionary of Computing Signs, NOEMA: a Multimedia Dictionary of GSL Basic Vocabulary and A Children's Dictionary of GSL) mainly for educational purposes, but complete decoding of the language structure is not yet publicly available.

The linguistic part of the project is based on overall assumptions for the adequacy of signed languages as by Stokoe [20] and Woll and Kyle [15], among many. Greek sign language is analyzed to its linear and non-linear (simultaneous) components [18][8]. The linear part of the language involves any sequences of lexical and functional tokens and their syntactic relations, while non-linear structures in GSL, as in all known sign languages, are present in all levels of the grammar. Each sign in GSL is described as to its handshape, location, movement, orientation, number of hands and use of any obligatory non-manually articulated elements (e.g. mouth patterns, head and shoulder movements, facial expression and other non-manual features), based on the Stokoe model.

In the project it was considered essential that the output is as close to native GSL as used in the Greek deaf community. In this respect, forms of 'signed Greek' or other manual codes for the teaching of Greek were excluded and the two languages (GSL and Greek) were treated as the first and second language respectively for the users of the platform, quite as other bilingual platforms may function outside the domain of special education.

3. LANGUAGE RESOURCES OF THE PROJECT

Implementation of both the tutoring and the summarization tools of the platform require collection of extensive electronic language resources for GSL as regards the lexicon and the structural rules of the language [6]. The actual data of the study are based on basic research on GSL analysis undertaken since 1999 as well as on experience gained by projects NOEMA and PROKLESE [7]. The data consist of digitized language productions of deaf native GSL signers and of the existing databases of bilingual GSL dictionaries, triangulated with the participation of deaf GSL signers in focus group discussions. The project follows methodological principles on data collection and analysis suitable to the minority status of GSL. Wherever the status of individual GSL signs is in consideration, the Greek Federation of the Deaf is advised upon, too.

Many of the grammar rules of GSL are derived from the analysis of a digital corpus that has been created by videotaping native signers in a discussion situation or when performing a narration. This procedure is required, because there exists little previous analysis of GSL and rule extractionhas to be based on actual data productions of native signers. The basic design of the system, except for the educational content this currently supports, focuses on the ability to generate sign phrases, which respect the GSL grammar rules in a degree of accuracy that allows them to be recognized by native signers as correct utterances of the language.

In this respect the SYNENNOESE project offers a great challenge for in-depth work on both directions, lexicography and linguistic analysis of GSL. For the first time, research goes beyond a mere collection of glosses and moves further from many previous bilingual dictionaries of sign languages [4], into the domain of productive lexicon [21], i.e. the possibility of building new GSL glosses following known structural rules, and also challenge automatic translation in predictable environments, using an effective module/interface for the matching of structural patterns between the written input and the signed output of the platform. It is a design prerequisite that the system of GSL description should have an open design, so that it may be easily extendible allowing additions of lemmas and more complicate rules, with the long term objective to create an environment for storage and maintenance of a complete computational grammar of GSL. From a linguistic point of view the resulting database of glosses, rules and tendencies of GSL will be a significant by-product of the project, of great value to future applications.

3.1 Grammar content definition

In the early implementation phase, the subsystem for the teaching of GSL grammar covered a restricted vocabulary and a core grammar capable of analyzing a restricted number of main GSL grammatical phenomena, which might be argued that belong to signing universals. Synthesis of GSL requires the analysis of the GSL signs into their phonological parts and their semantics. It was agreed that only monomorphemic signs that use only one handshape were to be initially analyzed, so that feedback from the technical team would determine further steps. In the second stage, more complicated sequential structures of signs are considered (e.g. compound word-signs) and once individual signs are transcribed and stored in a database, additional tiers such as basic non-manual features can be added without technical difficulties.

At the stage of grammatical analysis, findings from other sign language grammars, as well as the views of our deaf native user consultants are taken into account in order to verify findings. It is admitted that there is even more work to be done on the pragmatics of GSL and its relation with real-world situations (e.g. for the use of indexes or classifiers), and these are noted as future aims of the platform.

Furthermore, an interesting parameter of a virtual signer is the ability to sign letters of the written alphabet (fingerspelling). This technique is useful in cases of proper nouns, acronyms, terminology or general terms for which no specific sign exists. Fingerspelling is used extensively in some other sign languages such as the American Sign Language (ASL) or the British Sign Language (BSL); our evidence in GSL suggests that it is only used occasionally, rarely incorporating fingerspelled loans into the core of the language. From a technical point of view, however, it is quite simple for a VC to fingerspell as this process includes no syntax, movement in signing space or non-manual grammatical elements. Many previous attempts of sign animation would go up to the level of fingerspelling or signing only sequential structures of a representation of the written or spoken language. Since then technology has developed and so has linguistic description of sign language structures. On the other hand few deaf people in Greece use fingerspelling or a code such as 'Signed Exact Greek' extensively. For these reasons the present work aims to represent a form of GSL as close to natural fluent signing as possible, and only uses fingerspelling occasionally, for example in language games, where teaching of written Greek is the focus.

3.2 Notation and glossing

In order to decide on the notation to be followed for sign recording in the lexical resources database, the existing international systems of sign language recording were evaluated. Notation represents a vital part of the whole engine as it serves for the communication between the linguistic subsystem that determines the meaningful movements in the context of GSL and the technological subsystem that performs these movements with a synthetic 3D model signer.

Tools utilized for the transcription and notation include HamNoSys, a pictographic notation system developed by the University of Hamburg for the description of the phonology of signs [19]. This notation forms the corpus of GSL lemmas while for the representation of sequential structures, i.e. in the phrase level, the ELAN language annotator developed by the Max-Planck Institute of Psycholinguistics in Nijmegen, the Netherlands, will be used. We considered these two systems as most suitable to the text-to-sign animation according to reviews of recent relevant projects. The classic Stokoe model is used for the morpho-phonological description, with one additional tier with written Greek words of harsh semantic equivalents of utterances. An aim of the project is to add more tiers as the project continues, such as those mentioned above on the use of non-manual features and on pragmatics, using the existing symbols in HamNoSys and ELAN. Signwriting was another transcribing tool under consideration, but was not chosen, given the expected compatibility of HamNoSys within the Elan tiers in the near future.

4. TUTORING SYSTEM DESCRIPTION – CORPUS OF EDUCATIONAL MATERIAL

The test bed learning procedure concerns teaching of GSL grammar to early primary school pupils, whereas the platform also incorporates a subsystem that allows approach by the deaf learner to material available only in written Greek form by means of a signed summary. The learning process in practice will involve an initiator of the session, the students in groups or alone and a teacher-facilitator of the process, physically present with the students. The process can take place in real-time or can be relayed. There is provision of a virtual whiteboard, icon banks and chat board visible in the screen along with the virtual signer for common use in the classroom. The participants will also be able to see each other in real time through a web camera, in order to verify results of GSL learning.

Specifications for the formation of GSL resources of the application are crucially based on exhaustive research in the official, recently reformed, guidelines for the teaching of Greek language and of GSL in primary schools for the deaf. The educational content of the platform follows the same guidelines as the hearing children's curriculum, so that the same grammatical and semantic units can be taught in the two languages, GSL and spoken / written Greek. Concepts such as subject-object relations, types of verbs, discourse functions of the language form the units of the curriculum in the SYNENNOESE project so that the same principles are taught under the same platform, but without projecting a mirror image of the Greek grammar onto GSL. For the selection and arrangement of the educational material the project is in close cooperation with the Pedagogical Institute in Athens, which is the main official agency in charge of the development of educational material.

The first group of exercises deals with signs that use the same handshape but start from different positions with respect to the signer's body or the neutral signing space and consist of different movements. An example of such a group in GSL includes the words 'table', 'house', donkey', 'slipper' and 'tent'. In this framework, young pupils are initially presented with the VC signing each word in a particular group and a sketch depicting the same concept; the use of sketches instead of written words is adopted since very young pupils have not developed skills related with spoken or written languages and thus, their mother tongue is the relevant sign language. In the following, pupils go through a number of drills, similar to the ones found in usual language teaching classes. These drills consist of choosing the correct sketch relating to a random sign performed by the VC and matching different instances of the VC with the correct sketch, by picking from an on-screen sketch pool.

The second group of exercises includes signs with similar or semantically related meaning, signed with the same or different handshapes. An example is the group 'human', 'tall', 'fat', 'child', 'female'. The drills here are the same with the ones in the first exercise group, as is also the case with the third group of exercises. In this category, sign pairs are formed, consisting of signs composed of same phonological features (handshape, movement, location, palm orientation) but differing in their grammatical classification, e.g. 'sit-chair', 'eat-food' and 'love$_{verb}$-love$_{noun}$' by means of movement repetition.

5. TECHNICAL CONSIDERATIONS

The implementation team has reviewed currently available VC and animation technologies for the representation of sign language in order to adopt one of the most prominent technological solutions. The movements of a synthetic 3D signing model have to be recorded in a higher and reusable level of description, before they are transformed in parameters of body movement (such as Body Animation Parameters – BAPs according to the MPEG-4 model). In the area of text-to-sign animation there have been some similar projects (VISICAST, Thetos, SignSynth and eSIGN among them) that SYNENNOESE uses as background.

H-anim [11] is a set of specifications for description of human animation, based on body segments and connections. According to the standard, the human body consists of a number of segments (such as the forearm, hand and foot), which are connected to each other by joints (such as the elbow, wrist and ankle). As is mentioned in the standard description, the main goals of the h-anim standard are compatibility, flexibility and simplicity. In this framework, a human body is defined as a hierarchy of segments and articulated at joints; relative dimensions are proposed by the standard, but are not enforced, permitting the definition and animation of cartoon-like characters. In addition to this, different levels of skeleton articulation (Levels of Articulation – LOA) are available, catering for applications with different requirements: for example, a cartoon-like character and a martial arts computer game have inherently different needs for the flexibility of the relevant VC's body. Another welcome feature of the h-anim standard is that prominent feature points on the human body are defined in a consistent manner, via their names and actual locations in the skeleton definition. As a result, a script or application that animates an h-anim compatible VC is able to locate these points easily and concentrate on the high level appearance of the animation process, without having to worry about the actual 3D points or axes for the individual transformations. In the developed architecture, this is of utmost importance, because sign description is performed with respect to these prominent positions on and around the virtual signer's body.

For the recording and definition of handshape and gestures, motion tracking and haptic devices (such as CyberGrasp or Acceleration Sensing Glove with a virtual keyboard) were initially considered; however, it was agreed that, if the HamNoSys notation commands would provide acceptable quality, based on the initial implementation, motion capture sequences will not need to be applied. In any case, semantic notation is a far more flexible and reusable solution than video files or motion capture, since an h-anim VC can take advantage of the dynamic nature of phonological and syntactic rules.

5.1 Adopted 3D technologies

For the content designer to interact with a VC, a scripting language is required. In our implementation, we chose the STEP language (Scripting Technology for Embodied Persona) [10] as the intermediate level between the end user and the virtual actor. A major advantage of scripting languages such as STEP is that one can separate the description of the individual gestures and signs from the definition of the geometry and hierarchy of the VC; as a result, one may alter the definition of any action, without the need to re-model the virtual actor. The VC utilized here is compliant with the h-anim standard, so one can use any of the readily available or model a new one.

Scripted animation is an interchangeable and extensible alternative of animation based on motion capture techniques. One can think of the relation between these two approaches similarly to the one between synthetic animation and video-based instructions: motion capture can be extremely detailed with respect to the amount and depth of information, but is difficult to adjust or adapt when produced and typically requires huge amounts of storage space and transmission capacity to deliver. On the other hand, scripted animation usually requires manual intervention to compile and thus is minimal and abstract in the way it represents the various actions of the avatar. As a result, such scripts require a few hundred characters to describe and can be reused to produce different instances of similar shape [9]. This is illustrated in the code snippet in Figure 2, which illustrates the required transformations for the right hand to assume the 'd'-handshape. As is easily demonstrated, the same code of the left hand can be compiled by mirroring the described motion, while other, more complicated handshapes can start with this representation and merely introduce the extra components into it.

```
par([
   turn(humanoid,r_thumb1,rotation(1.9,1,1.4,0.6),
   very_fast),
   turn(humanoid,r_thumb2,rotation(1,0.4,2.2,0.8),
   very_fast),
   turn(humanoid,r_thumb3,rotation(1.4,0,0.2,0.4),
   very_fast),
   turn(humanoid,r_index1, rotation(0,0,0,0),
   very_fast),
   turn(humanoid,r_index2,rotation(0,0,0,0),
   very_fast),
   turn(humanoid,r_index3,rotation(0,0,0,0),
   very_fast),
   turn(humanoid,r_middle1,rotation(0,0,1,1.5999),
   very_fast),
   turn(humanoid,r_middle2,rotation(0,0,1,1.5999),
   very_fast),
   turn(humanoid,r_middle3,rotation(0,0,1,1.5999),
   very_fast),
   turn(humanoid,r_ring1,rotation(0,0,1,1.7999),
   very_fast),
   turn(humanoid,r_ring2,rotation(0,0,1,1.5999),
   very_fast),
   turn(humanoid,r_ring3,rotation(0,0,1,0.6000),
   very_fast),
   turn(humanoid,r_pinky1,rotation(0,0,1,1.9998),
   very_fast),
   turn(humanoid,r_pinky2,rotation(0,0,1,1.5999),
   very_fast),
   turn(humanoid,r_pinky3,rotation(0,0,1,0.7998),
   very_fast)
])
```

Figure 2: STEP code for a handshape

In the SYNENNOESE project, a syntactic parser decodes the structural patterns of written Greek and matches them into their equivalents in GSL [3]. These are fed into an automated system that decodes HamNoSys notation sequences for each lemma; this system essentially transforms single or combined HamNoSys symbols to sequences of scripted commands. A typical HamNoSys notation sequence consists of symbols describing the starting point configuration of a sign and the action that the signing consists of. Symbols describing the initial configuration refer to the handshape that is used during the sign and the starting position and orientation of the hand that performs the sign; if the other hand takes part in the sign, as is the case in the GSL version of 'doctor', it is the relative position of the two hands that matters, for example ‘the main hand touches the elbow of the secondary arm’. Other information includes symmetry, if both hands follow the same movement pattern and any non-manual components. Figure 3 shows a frame of the signing sequence for ‘donkey’; the VC shown here is ‘yt’, by Matthew T. Beitler, available at http://www.cis.upenn.edu/~beitler. A demonstration with limited vocabulary and some phrase examples can be found online at http://www.image.ece.ntua.gr/~gcari/gslv.

Figure 3: An instance of ‘yt’ signing ‘donkey’

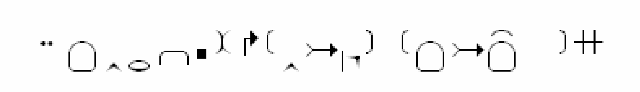


Figure 4: The HamNoSys sequence for the GSL version for ‘donkey’

Figure 5: The GSL version of ‘child’

Figure 4 shows the HamNoSys sequence for the particular sign, shown on the top of the page of the user interface. The first symbol here indicates that both hands perform the same movement, starting from symmetrical initial locations with respect to the signer’s torso. The second symbol indicates the handshape, which here is an open palm, referred to as the ‘d’-handshape in GSL, while the next shows palm orientation. The following symbols handle the starting position of the palm, which here almost touches the temple of the signer’s head. Symbols contained in parentheses describe composite movements, while the last character forces the signer to repeat the described movement.

```
par([
   turn(humanoid,r_elbow,
   rotation(-0.722,0.2565,0.1206,1.5760),fast),
   turn(humanoid,r_shoulder,
   rotation(-0.722,0.2565,0.1206,0.0477),fast),
   turn(humanoid,r_wrist,
   rotation(0,1.5,-1,1.570),fast)
]),
sleep(500),
par([
   turn(humanoid,r_shoulder,
   rotation(-0.598,0.2199,0.1743,0.0812),fast),
   turn(humanoid,r_elbow,
   rotation(-0.598,0.2199,0.1743,1.2092),fast)
])
```

Figure 6: The STEP code for the sign ‘child’

Figure 7: The GSL version of ‘children’

Figure 5 shows the VC signing the GSL version of ‘child’, while Figure 7 shows an instance for the sign for ‘children’. The design of the automated script production system enables us to use the description of the former sign (Figure 6) to construct the definition of its plural form. In this case, the plural form is shown by repeating the same downward hand movement, while moving the hand slightly to the signer’s right; direction is indicated by the symbol preceding the parenthesis, while its content describes this secondary movement. As a result, it is only necessary for the parser to indicate the particular modification of the initial sign required to produce the plural form of the lemma. In GSL, these forms are limited, thus enabling us to come up with efficient production rules, such as the one described above. Another possibility is to change the handshape for a sign, especially when the signer wants to indicate a particular quantity or number. Figure 8 shows the VC signing the GSL version of ‘day’, while Figure 9 shows the GSL version of ‘two days’: the difference here is that in the latter case the VC uses a two-finger handshape, instead of the straight-index finger handshape, to perform the same movement, starting from the same initial position. This difference is more evident in Figure 10, which shows the VC in a frontal view; this is actually a nice feature of the Blaxxun Contact 5 [2], the VRML plug-in shown in these figures. Despite the default tilted view being the one of choice from the part of the users, the ability to show frontal and side view of the sign is crucial in learning environments, since it caters for displaying the differences between similar signs and bring out the spatial characteristics of the sign [14][13].

Figure 8: The GSL version of 'day'

Figure 9: The GSL version of 'two days'

Figure 10: The frontal view of the GSL version of 'two days'

6. IMPLICATIONS AND EXTENSIBILITY OF THE EDUCATIONAL PLATFORM

As an educational tool above all, the SYNENNOESE platform offers a user-friendly environment for young deaf pupils aged 6 to 9, so they can have visual translation of words and phrases. The signed feedback acts as a motivating tool for spelling Greek words and structuring sentences correctly, as well for evaluating one's performance. For deaf young students as a group with special needs, the platform draws some of the accessibility barriers, and the possibility of home use even makes it accessible to family, thus encouraging communication in GSL, but also access to the majority (Greek) language. New written texts can be launched, so SYNENNOESE may receive unlimited educational content besides primary school grammar units. On the other hand, unlimited school units, such as the increasing special units with individual deaf students in remote areas can link with one another via the SYNENNOESE platform.

Moreover, text-to-sign translation can be extended and applied to different environments such as Greek language teaching to deaf students of higher grades, GSL teaching for hearing students, Greek for specific purposes such as to adult literacy classes for the Deaf etc. In this context, more domains of GSL grammar can be described and decoded, making the output closer to natural signed utterances as our analysis proceeds. This is a challenge not only for theoretical research, but also for computer science and applied linguistic research.

Furthermore, a database with the bulk of GSL utterances, described as to their features from the phonological up to the pragmatic level will be the major outcome of the whole project. In this way the representation of GSL structures can be matched to the equivalent ones of written Greek, and it will be a challenge to be able to compare directly the grammars of the two languages. In much the same way structures of GSL will easily be compared with counterparts from ASL or BSL [4] for research across signed languages.

7. PROBLEMS AND LIMITATIONS

The main limitations of the study are described below. These are divided into linguistic, educational and technical ones. Most of the limitations are typical to sign animation projects, and they were expected before the beginning of the project.

Regarding the linguistic and educational aspects of the project, one of the major issues that needs to be addressed is the fact that in some areas of the language there are no standardized signs, so there may be some theoretical objections as to the use of particular entries. However, a platform such as the one described allows for multiple translations and does not have any limitations as to the size of files, which was the case, for example in previous DVD-based GSL dictionaries, containing video entries. Moreover, the platform will be open to updates through the script authoring process.

Another issue is the choice of entries to be included in each stage of the platform development depending on the complexity of their phonological characteristics. As mentioned already in the section on grammar content definition, monomorphemic entries were agreed to be included in the first stage. In the next stages there is gradual provision for polymorphemic signs, compound signs, functional morphemes, syntactic use of non-manual elements, sequential and lastly simultaneous constructions of separate lexical signs, each stage to correspond with the level of linguistic research in GSL.

Besides this, the data available in GSL, when compared with data from written Greek, for example, are dauntingly scarce. Error correction mechanisms were sought after in order to assure reliability of results. Such back-up mechanisms are the use of approved dictionaries, the consultancy of Pedagogical Institute and the feedback from the Deaf Community, along with the continuing data from GSL linguistic research.

The most important technical problems include a solution for smooth transition between concurrent signs and fusion between handshapes so that neighboring signs in a sentence appear as naturally articulated as possible. In the context of the SYNENNOESE project, this issue has been tackled using a nice feature of the STEP engine, which at any time can return the setup of the kinematic chain for each arm. As a result, when the sign that is next in a sequence begins, the kinematic chain is transformed to the required position without having to take into account its setup in the final position of the previous sign. In general, this would be problematic in general purpose animation, since the h-anim standard itself does not impose any kinematic constraints; thus, random motion might result in physiologically impossible, puppet-like animation. In the case of signing though, almost all action takes place in the signing space in front of the signer and starting from the head down to the abdomen; in this context, there are no abrupt changes in the chain setup.

Another issue regarding the animation representation has to do with circular or wavy movement. Since the description follows the same concepts as keyframed motion, circular movement or generally, paths following a curve must be approximated with discrete key positions. This often results in losing the relative position of the hands, as shown in Figure 11, which depicts the final position for the sign ‘boat’; this sign should normally end up with palms touching, but since this process is designed with the position on the palm in mind, keeping hands together is not a straightforward task.

Figure 11: Problems in the GSL version of ‘boat’

In addition to this, a major factor in sign synthesis is the grammatical use of non-verbal signs, such as meaningful or spontaneous facial expression [12] and eye gaze, particularly when eye gaze has to follow the track of hand movements. Similar problems are anticipated on mouth movements on prosodic features of sign phonology. Mouthing the visible part of spoken Greek words will not be an issue for the project yet, but this, too is anticipated as a problem to deal with in the future, as all of the above non manually signed features are considered as internalized parts of GSL grammar. At the moment, the only possible non-manual sign components possible to animate with the STEP platform are gazing towards the signer’s moving hands and forward torso leaning, in the case of asking a question. In general, the STEP engine does not yet feature facial animation, so the project team is considering moving to a pure MPEG-4 [17] based platform. A nice example of maturing MPEG-4 synthetic technology is the VC named ‘Greta’ [5] which supports all required manual and non-manual components, including visemes, the visual counterpart of phonemes used for lip-reading, high-level facial expression, e.g. ‘surprise’ associated with an exclamation mark or simple facial and head movement, such as raising the eyebrows or tilting the head upwards to indicate negation (see Figure 12).

Figure 12: Greta displaying manual and non-manual signs

The ultimate challenge, as in all similar projects, remains the automatic translation of the language. It is still too difficult to produce acceptable sentences in the automatic translation of any language at the moment, even more so a minor, less researched language with no written tradition such as GSL. Realistically the teams involved in the SYNENNOESE project can expect as an optimum result the successful use of automatic translation mechanisms in GSL only in a restricted, sub-language oriented environment with predetermined semantic and syntactic characteristics.

8. CONCLUSION

In this paper we have described the underlying design principles and implementation of a web-based virtual signer software component, utilizing language resources suitable for young pupils. This component uses standard linguistic and virtual character technologies to provide semantic and syntactic information from written text and encode it with reusable and extensible sign notation representations. These representations are readable by the VC platform, making them suitable for teaching GSL and providing signed summaries of documents.

9. ACKNOWLEDGMENTS

The authors wish to acknowledge the assistance of all colleagues involved in the design and implementation of the discussed platform and the representatives of the Greek Deaf Community. This work was partially funded by the national project grant SYNENNOESE (GSRT: e-Learning 44).

10. REFERENCES

[1] Bellugi, U. and Fischer, S., A comparison of Sign language and spoken language, *Cognition*, 1, 173-200, 1972

[2] Blaxxun Contact 5, http://www.blaxxun.com/en/products/contact/

[3] Boutsis S., Prokopidis P., Giouli V. & Piperidis S., Robust Parser for Unrestricted Greek Text. *Proceedings of the 2nd Language Resources and Evaluation Conference*, 467-474, 2000.

[4] Brien D. and Brennan, M, *Dictionary of British Sign Language / English*. Faber and Faber, Boston, 1992.

[5] De Rosis, F., Pelachaud, C., Poggi, I., Carofiglio, V. and De Carolis, B. , From Greta's mind to her face: modeling the dynamics of affective states in a conversational embodied agent, International Journal of Human-Computer Studies, Academic Press, Vol. 59, Issue 1-2, pp. 81 – 118, 2003.

[6] Efthimiou, E., Sapountzaki, G., Karpouzis, K., Fotinea, S-E. Developing an e-Learning platform for the Greek Sign Language, in K. Miesenberger, J. Klaus, W. Zagler (Eds.), *Computer Helping People with Special Needs, Lecture Notes in Computer Science,* Vol. 3118, pp. 1107 - 1113, Springer-Verlag, 2004.

[7] Efthimiou, E., Vacalopoulou, A., Fotinea, S.-E., Steinhauer, G., Multipurpose Design and Creation of GSL Dictionaries, *Proceedings of Workshop on Representation and Processing of Sign Languages, LREC 2004*, Lisbon, May 2004, pp. 51-58.

[8] Engberg – Pedersen, E., *Space in Danish Sign Language*, Signum Press, Hamburg, 1993.

[9] Foulds, R., Biomechanical and Perceptual Constraints on the Bandwidth Requirements of Sign Language, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, March 2004, pp. 65–72.

[10] Huang, Z., Eliens, A., and Visser, C., STEP: A Scripting Language for Embodied Agents, *Proceedings of the Workshop on Lifelike Animated Agents*, 2002.

[11] ISO/IEC 19774 — Humanoid Animation (H-Anim), http://www.h-anim.org

[12] Karpouzis, K., Raouzaiou, A., Kollias, S., "'Moving' avatars: emotion synthesis in virtual worlds". C. Stephanidis, J. Jacko (eds.), *Human – Computer Interaction: Theory and Practice*. Vol. 2. 503 – 507. Lawrence Erlbaum Associates, Inc, 2003.

[13] Kennaway, R., Experience with, and Requirements for, a Gesture Description Language for Synthetic Animation. *5th International Workshop on Gesture and Sign Language based Human-Computer Interaction*, Genova, 2003.

[14] Kennaway, R., Synthetic Animation of Deaf Signing Gestures. *International Gesture Workshop*, City University, London, 2001.

[15] Kyle, J. G. and Woll, B., *Sign Language: the study of deaf people and their language*, Cambridge University Press, 1985.

[16] Liddell, S. *American Sign Language Syntax*, The Hague: Mouton, 1980.

[17] MPEG-4 Home page, http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm

[18] Padden, C., The Relationship between Space and Grammar in American Sign Language Verb Morphology, *Theoretical Issues in Sign Language Research*, Gallaudet University Press, 1989.

[19] Prillwitz et al, *HamNoSys. Version 2.0. Hamburg Notation System for Sign Language. An Introductory Guide*, 1989.

[20] Stokoe W and Kuschel R., *For Sign Language Research*, Linstock Press, 1978.

[21] Wilcox, S., Scheibmann, J., Wood, D., Cokely, D. & Stokoe, W., Multimedia dictionary of American Sign Language. *Proceedings of ASSETS Conference*, pp. 9-16, 1994.

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.

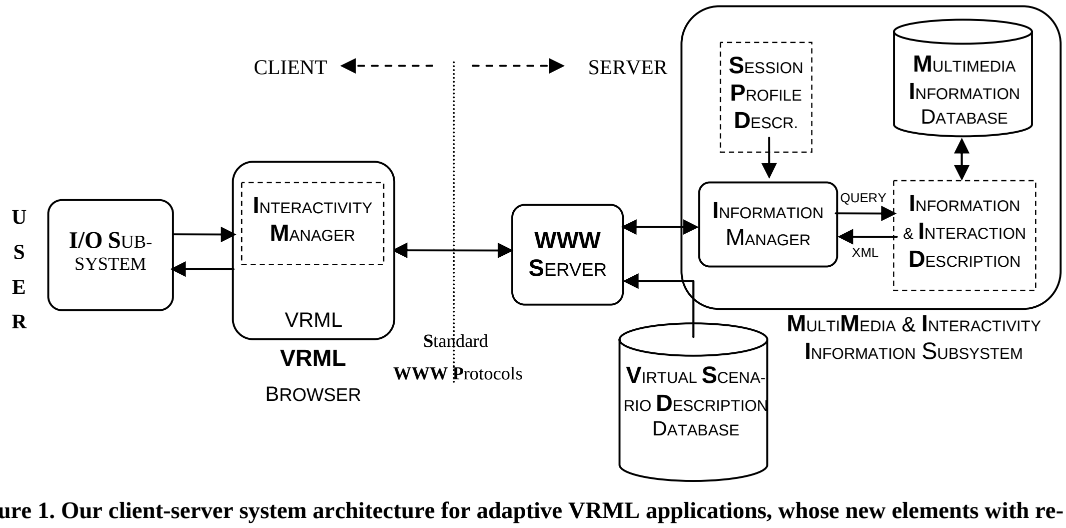


Figure 1. Our client-server system architecture for adaptive VRML applications, whose new elements with respect to the standard one are the *Information manager* and the *Information and Interaction description* in the server, and the *Interactivity manager* in the client.

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 hard-wired 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
<?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.

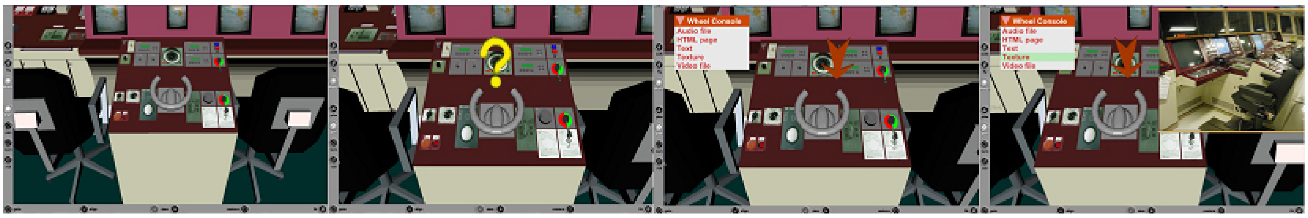
4.3 Interactivity Manager

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]).

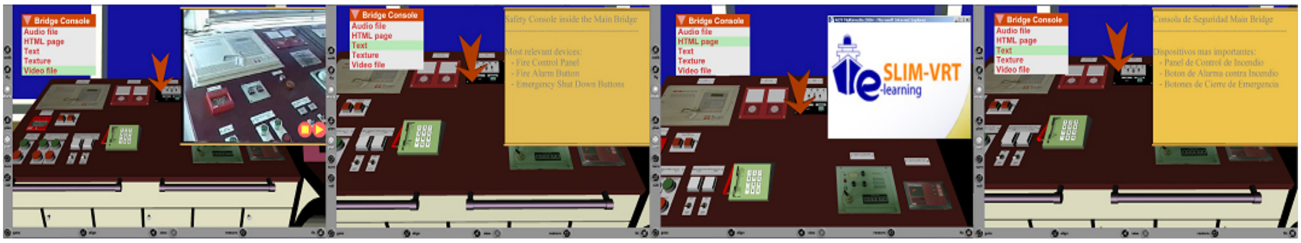
2. Next, the Java class executes the SAX parser to access the parameters stored in the *model controller* XML file as the result of the filtering process.

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.

Employing Virtual Humans for Education and Training in X3D/VRML Worlds

Lucio Ieronutti
HCI Lab, Dept. of Math and Computer Science,
University of Udine,
via delle Scienze 206, 33100 Udine, Italy
ieronutt@dimi.uniud.it

ABSTRACT
Advances in computer graphics, improvements in hardware performance, and network technologies have enabled a new class of interactive applications involving *virtual humans*, three-dimensional simulations of human beings. Virtual humans are more and more used in multimedia learning environments, e.g. to explain maintenance operations, to train medical staff, or can be employed as virtual teachers. Nevertheless, in Web3D sites virtual humans are in practice rarely used, since they are complex to implement and their proper development requires knowledge of several disciplines (e.g., biomechanics, kinematics, artificial intelligence, computer graphics,…). Moreover, the programming effort to develop and re-use the virtual human into different learning environments can be considerable. In this paper, we propose a general architecture that allows Web3D content creators to easily integrate virtual humans into learning environments. The proposed solution can be used independently from the specific learning application domain (e.g. from a technical presentation to an history lesson). To test the applicability and effectiveness of our approach, we have applied it in a virtual museum of computer science.

Categories and Subject Descriptors
I.3.6 [**Computer Graphics**]: Methodology and Techniques – *interaction techniques.* I.3.7 [**Computer Graphics**]: Three dimensional Graphics and Realism – *Virtual Reality.* H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information System – *Artificial, augmented, and virtual realities.*

General Terms
Design, Experimentation, Human Factors.

Keywords
Virtual Environment, Learning Environment, Virtual Humans, Embodied Agents, H-Anim.

1. INTRODUCTION
Virtual humans, i.e. three-dimensional simulations of human beings, are more and more used for different applications in multimedia learning environments: i) they are used to explain physical and procedural human tasks (e.g. maintenance operations) [1][13] by allowing users to be given less theoretical explanations which are more intuitive and engaging; ii) they are employed in medicine, from training applications designed to train civilian officers to recognize and interact with mentally ill people [8], to applications that range from simulations of emergencies in first aid [3] to simulations of surgical operations [4]; iii) virtual humans are used in military applications to train servicemen [15], by allowing one to reproduce very dangerous situations; iv) they are used as virtual personal teachers that complement their explanation by referring to virtual objects and places, in order to simplify the understanding of the lesson. The virtual teacher can lead users through the environment [5] and present different topics by following a logical order.

In general, virtual humans employ locomotion abilities and perform actions (such as deictic gestures) to focus students' attention on most important aspects and interact with virtual objects, while they can use facial expressions [14] in order to make the communication with users more realistic, effective and engaging. Virtual humans can communicate with users by exploiting both verbal and nonverbal communication; a virtual human can use gaze and gestures to focus the student's attention [11][12][13] during its explanation. For example, it can use gaze to regulate turn-taking in a mixed-initiative dialogue [3], while head nods and facial expressions can provide unobtrusive feedback on the student's utterances and actions without disrupting the student's train. Moreover, the presence of an anthropomorphic agent may increase the student's arousal and motivation [10]. From a human-computer interaction point of view, the interaction with virtual humans is based on metaphors consistent with the real world experience of users and then it suggests the possibility of more natural ways of communication.

There is a large amount of research in the field of virtual humans, and a large number of different approaches have been proposed in literature; each of them varies in appearance, function and autonomy according to the application field and the required detail and accuracy. Nevertheless, virtual humans are rarely used in Web3D sites, since they are complex to implement and there are no tools for supporting the development of virtual humans with complex behaviors in these Web sites. As a result, since the Web3D content creator has to implement virtual humans mostly by hand, and the programming effort to develop and re-use the virtual human into different learning environments can be considerable.

In this paper, we propose a general architecture for H-Anim characters [7], called Virtual Human Architecture (VHA), that allows Web3D content creators to easily employ virtual humans in Web-based applications for learning, education and training. To test benefits of our approach, we use it in a 3D Web site representing a Computer Science museum in which the virtual human leads users through the environment, it invites students to interact with devices and it provides technical explanations regarding the functioning of different devices.

This paper is structured as follows. In Section 2, we present problems related to the development of virtual humans and discuss the lack of supports for implementing virtual humans in Web3D sites. In Section 3, we present the Virtual Human Architecture, a general architecture that allows content creators to develop virtual humans in Web3D sites without worrying about low-level implementation aspects. In Section 4, we show how the proposed architecture can be used in a multimedia learning environment by providing a practical usage example. Finally, in Section 5, we discuss current limitations of the proposed approach and outline how we plan to overcome them.

2. MODELING VIRTUAL HUMANS

The development of virtual humans requires to acquire the knowledge of different disciplines, such as computational geometry, kinematics, artificial intelligence, computer graphics, and bio-mechanics. The complexity of building an embodied agent requires to subdivide the problem; this can be done in a hierarchical way [6], as shown in Figure 1.

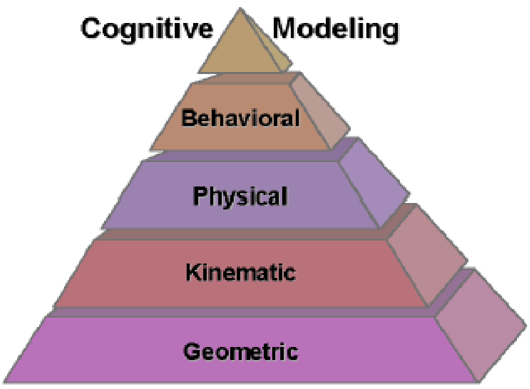


Figure 1– The modeling hierarchy proposed by [6]

At the base of the pyramid there is the *geometric* layer that concerns the definitions of the virtual human model and its appearance. In the *kinematic layer*, the virtual human is represented as a set of rigid bodies, called *segments*, organized hierarchically and connected by *joints*. From this point of view, an animation can be defined in two ways: i) by specifying joints rotations, or ii) by defining (or automatically computing) positions of segments extremities in time. The latter approach uses inverse kinematics, technique that allows one to compute the joints configuration (in terms of rotation values) needed to reach the specified position; this technique is especially suitable to control hands and feet movements. In the *physical layer* the animation is obtained by applying physical laws to different parts of the body; this method is used to compute complex animations, such as skin deformations or the hairs movement. The *behavioral layer* represents the instinctive behavior of the virtual human (e.g. in terms of stimulus-action associations). *The cognitive modeling* instead binds various stimuli with reasoning process that allow the virtual human to search for the most suitable reaction; cognitive models go beyond behavioral models in that they govern what the virtual human knows, how that knowledge is acquired, and how it can be used to plan actions.

2.1 Modeling Virtual Humans in X3D/VRML

Although to implement virtual humans on the Web different technologies and approaches can be employed (see [9] for an overview), we focus our attention on H-Anim [7], the standard for implementing humanoids by using X3D/VRML technologies.

H-Anim defines the virtual human body as a set of *segments* organized hierarchically and connected by *joints*; each joint is defined by its position and its own rotation value. A humanoid animation is defined in X3D/VRML by specifying different rotation values of joints in time; the resulting motion is generated by smoothly interpolating specified rotation values. Moreover, since H-Anim defines the name of different joints, it allows one to apply the same animation to different virtual humans. This kind of animations are called *pre-stored animations*, since the complete description of the movement is specified in advance. Actually H-Anim standard supports only this kind of animations.

To animate a virtual human it is more convenient to use another kind of animation, called *parametrized animations*, that use a small set of parameters to generate at execution-time an animation as a function of these parameters. Parametrized animations are more general and flexible than pre-stored ones, since they can generate a variety of movements by changing animation parameters. Usually parametrized animations use inverse kinematics to control end-effectors movements (e.g. feet and hands) and employ path planning algorithms to generate collision-free motions. A typical example of parametrized animation is the walking motion; by starting from an high-level description of the movement (e.g. specified the initial and final humanoid position and by defining the set of parameters that characterizes the movement, such as the length of a single step), the corresponding animation is generated by using a path planning algorithm to compute a collision-free trajectory, and by employing inverse kinematics to derive a legs movement that avoids compenetration with the walking surface. Unfortunately H-Anim does not support parametrized animations.

Moreover, H-Anim does not specify the way for describing the high-level behavior of the virtual human. As a result, the implementation of an animated virtual human in learning and training environments is a trial-and-error, time-consuming activity for the Web3D content creator.

3. The proposed VIRTUAL HUMAN ARCHITECTURE (VHA)

In this section we briefly describe the high-level architecture we propose (illustrated in Figure 1). The main internal module of the architecture are i) the *Behavioral Engine* and ii) the *Animation Engine*. It is important to note that the proposed architecture is a good compromise between the required realism (or at least believability) of the representation (at each level of the modeling hierarchy [6]) and the efficiency of the simulation (the simulation has to be carried out in real-time on common home computers).

The Behavioral Engine, by sensing user's interactions and depending on the defined behavior of the virtual human, identifies both the information that has to be provided to the user and animations that the virtual human has to perform.

Information are taken from a database and are at the same time both displayed on a semi-transparent On Screen Display (OSD for short, see Figure 3) and presented by using a synthesized voice (e.g. using the Microsoft Text-to-Speech engine).

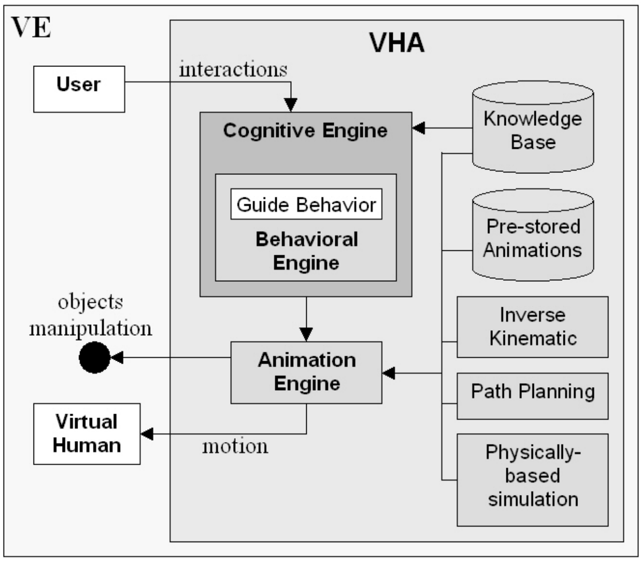


Figure 2 – The Virtual Human Architecture (VHA)

By using the inverse kinematics, path planning algorithms, pre-store animations and by exploiting objects information stored into a database (e.g. their location and orientation into the virtual environment), the Animation Engine generates at run-time the required animation.

In the following we describe the two modules in more detail.

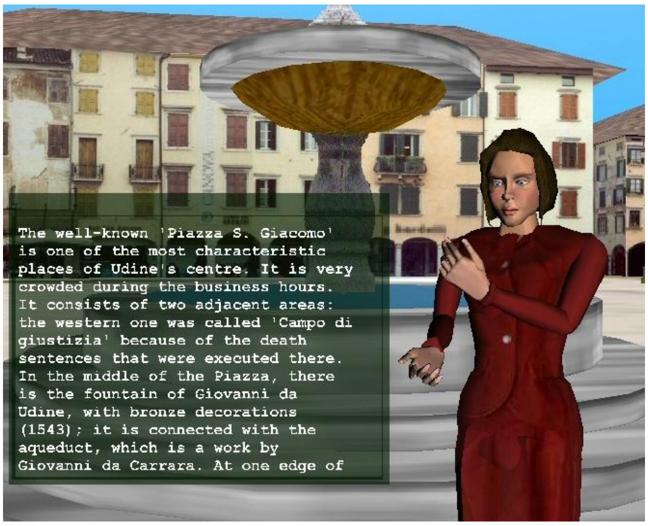


Figure 3 – The required presentation text is read by a text-to-speech engine and displayed on a OSD.

3.1 Animation Engine

The Animation engine allows the virtual human to perform parametrized animations such as walking, stairs-climbing, grasping, and the ability to make facial expressions.

The walking animation is computed by using inverse kinematic techniques; this approach allows the VHA to generate realistic and flexible walking animations. Given the position of footprints on the desired path, VHA automatically creates the walking animation by computing trajectories of the humanoid pelvis and feet. Since VHA takes into account both the virtual human model and the morphology of the walking surface to generate the animation, the Animation Engine is able also to generate stair-climbing animations only by specifying the walking surface parameters.

VHA adopts inverse kinematic techniques also to generate grasping animations; these animations can be used for example in virtual training environments to demonstrate how to repair a broken engine or to explain how to carry out maintenance procedures.

Moreover, VHA supports expressive virtual humans (humanoids that are able to display facial expressions); from a Human-Computer Interaction point of view, interacting with an expressive virtual human makes the conversional more believable, realistic and engaging for users. Among different techniques proposed in literature for this purpose, the proposed architecture uses the physics-based one, i.e. technique that controls facial expressions by acting on muscles contraction and computing the resulting face expression according to physic laws (e.g. using the mass-spring-model).

3.2 Behavioral Engine

Models of virtual human behavior are integrated into the architecture; these models specify how the virtual human responds to user input in terms of actions performed and information presented. Each model of the virtual human behavior is represented by a Finite-State Machine (FSM for short); this structure specifies how the virtual human acts given the current state and considered student interactions.

Each FSM is represented with the FSM G = (V, E), where V are the set of nodes n_i , while E are the set of oriented edges (n_i, n_j). Each node corresponds to a particular state of the virtual human, while each edge corresponds to a transition that allows the virtual human to change its internal state. Each transition (n_i, n_j) is characterized by the couple conditions-actions (c_{ij}, a_{ij}): c_{ij} is the set of conditions that determine the applicability of the transition, while a_{ij} is the set of actions that the virtual human performs if the corresponding transition is activated by the Behavioral Engine. A transition can be activated if and only if corresponding conditions are satisfied.

Given a current state of the virtual human, the Behavioral Engine senses user's interactions and determines what conditions are satisfied, identifies applicable transitions, activates one of them and returns the set of actions associated to the chosen transition.

Each FSM represents a different behavior and, in a didactical application, can correspond to the structure of an interactive lesson. From this point of view, nodes can represent different concepts (or group of concepts), while edges can represent relations between concepts. This way, given the concept c_k (that is associated to the state n_k), all nodes belonging paths connecting the initial state with n_k can be considered concepts necessary for the understanding of c_k. As a result, the user can actively participate to the lesson by influencing the order in which different concepts are presented, since the user behavior (e.g. its interaction with objects and the virtual human) determines the way in which the graph is explored.

4. An example of a learning environment: a 3D COMPUTER SCIENCE MUSEUM

The proposed architecture can be used in different learning environments independently from the application domain. To test the effectiveness of our solution, we considered two different case studies. First, we employed a virtual human into a learning environment aimed to explain the functioning of computer devices of the 70's. Second, we used the same virtual human as a guide into an architectural virtual reconstruction of cultural heritage. Although the considered Web3D sites differ in purposes and contents, only few modifications have been required to shift from the first to the second application. While in the first learning environment the virtual human provides technical information by demonstrating how different devices worked, in the second application the same virtual human is used for the promotion of cultural heritage, since its main function is to tell the history of different buildings by highlighting main architectural differences.

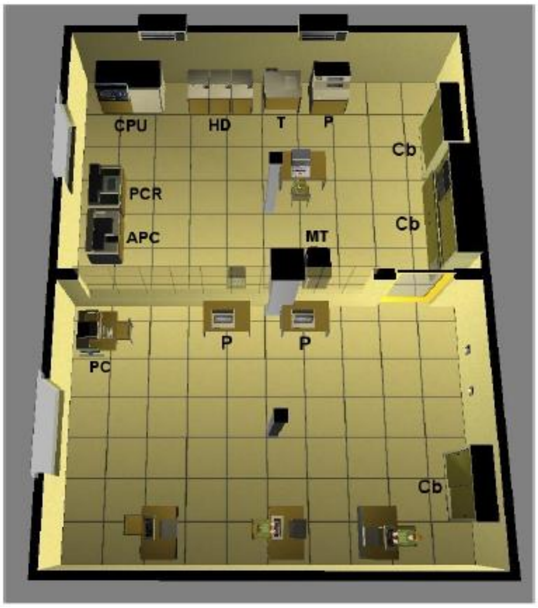


Figure 4 – Top view of 3D Computer Science Museum, with the different devices on display

Due to the limited available space, in this paper we focus only on the first mentioned learning environment. A screenshot of the second learning environment is provided in Figure 3.

The 3D Computer Science museum is based on the virtual reconstruction (developed using VRML and Java) of a typical data processing centre of the '70s, reproducing hardware from the Univac/Sperry 90/30 line. The main pedagogical goals for this virtual museum are concerned with pointing out the large differences between data processing centers in the '70s and current computers, e.g. by illustrating the mainframe – terminals architecture and the interaction based on text video terminals or (more often) punch cards.

The virtual data processing center is divided into two main rooms (as shown in Figure 4): a computer room, containing the main system, devoted to data processing under the control of technical staff, and a terminal room, containing punch card units and video terminals, devoted to activities that are preparatory to real data processing. Museum visitors have the possibility to: i) observe the different devices in their original context of operation; ii) obtain information on the devices, by clicking on them and reading and/or listening to a description of their features and functioning; iii) manipulate the devices to observe their internal parts. For example, the user can open cabinet units to examine their internal details and working (Figure 5 illustrates the case of hard disks).

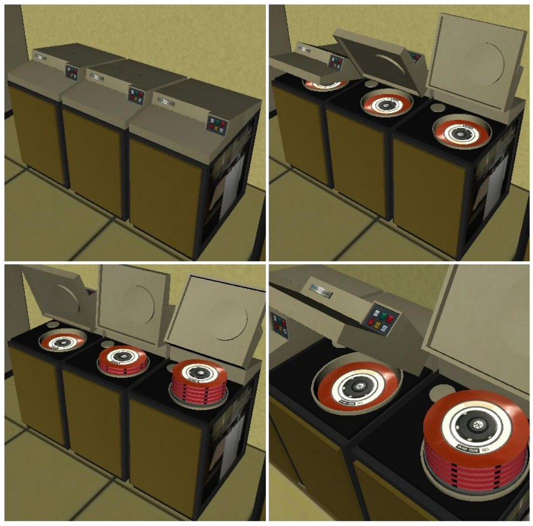


Figure 5 - Interactive components: hard disks

The virtual human leads student through the environment, it presents and describes different devices by following a logic order and by considering what information has been already presented to the user during the visit. If needed, the virtual human provides additional information (see Figure 6), provides comparisons between different devices (e.g. punch card and card reader), by organizing logically the lesson, e.g. the virtual human explains what memory devices were used in the 70's before providing the detailed punch card description.

To increase the realism of the user experience, we added the necessary furniture and included typical working people. Moreover, the audio channel is used to add typical noise and sounds of objects and human actions (for instance, printers, operator's typing, etc).

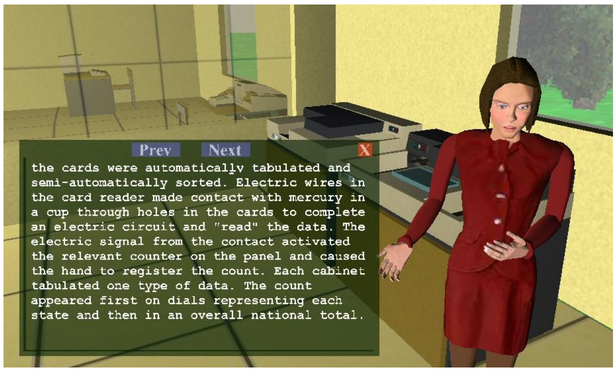


Figure 6– The virtual human while it is explaining the functioning of the card punch

We define different FSM that correspond to different structured information, each one designed to highlight a different aspect of the Computer Science Museum (e.g. an high-level introduction to the overall environment, an explanation of the hardware architecture, a description of work activities). The user can choose the lesson he intends to follow before the visit.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented a general architecture to develop virtual humans in Web3D sites. In particular, the architecture is used for learning and training applications in which virtual humans are used as virtual teacher and assistants. The proposed solution allows Web3D content creators to implement learning environments in different application domains. We have also presented a practical example that shows how virtual humans can be used to teach technical topics by using a 3D Computer Science Museum as a case of study.

We plan to extend the proposed approach by considering the user model in order to take into account needs, preferences and the individual knowledge of users. From this point of view, we plan to integrate our architecture with a tutoring system that dynamically generates Web content: this way, it is possible to provide users with personalized information, allowing the virtual human to present tailored lessons to different students. Users' information could be acquired by using an initial form that asks for typical information, exploiting a stereotypical knowledge, and by dynamically updating the student's profile by considering her interaction both with the virtual human and the environment.

Moreover, we intend to make the user-virtual human interaction more natural and intuitive; to achieve this goal, we plan to integrate into the proposed architecture a speech recognition engine; this solution allows users to communicate with virtual humans by using a well-known metaphor.

6. ACKNOWLEDGMENTS

This work is partially supported by the MIUR COFIN 2003 program and by the Friuli Venezia Giulia region (Regional Law 3/98, project "3D Web Sites for the Promotion of Tourism and Cultural Heritage").

7. REFERENCES

[1] Badler, N., Erignac, C., Liu, Y. Virtual humans for validating maintenance procedures. *Communications of the ACM*, 2002, 45(7), pp. 56-63.

[2] Badler, N., Webber, B., Clarke, J., Chi, D., Hollick, M., Foster, N., Kokkevis, E., Ogunyemi, O., Metaxas, D., Kaye, J., and Bindiganavale, R. MediSim: simulated medical corpsmen and casualties for medical forces planning and training. In *Proceedings of the National Forum: Military Telemedicine On-Line Today Research, Practice, and Opportunities*, IEEE Computer Society Press, 1996, pp. 21-28.

[3] Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S., and Stone, M. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of SIGGRAPH '94*, ACM Press, New York, 1994, pp. 413-420.

[4] Cavazza, M., and Simo, A. A virtual patient based on qualitative simulation. In *Proceedings of the 8th international conference on Intelligent user interfaces*, ACM Press, New York, 2003, pp. 19-25.

[5] Chittaro L., Ranon R., and Ieronutti L. Guiding Visitors of Web3D Worlds through Automatically Generated Tours. In *Proceedings of Web3D 2003: 8th International Conference on 3D Web Technology*, ACM Press, New York, 2003, pp. 27-38.

[6] Funge, J., Tu, X., and Terzopoulos, D. Cognitive Modeling: Knowledge, Reasoning, and Planning for Intelligent Characters. In *Proceedings of SIGGRAPH '99*, ACM Press, New York, 1999, pp. 29-38.

[7] H-Anim web site: http://www.h-anim.org/.

[8] Hubal, R. C., Frank, G. A., and Guinn, C. I. Lessons Learned in Modeling Schizophrenic and Depressed Responsive Virtual Humans for Training. In *Proceedings of the Intelligent User Interface Conference*, ACM Press, New York, 2003, pp. 85-92.

[9] Lau, R.W.H., Li, F., Kunii, T.L., Guo, B., Zhang, B., Magnenat-Thalmann, N., Gutierrez, M. *Emerging Web Graphics Standards and Technologies*, IEEE Computer Graphics and Applications, 2003, pp. 66-75.

[10] Lester, J. C.,Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., and Bhogal, R. S. The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI '97*, ACM Press, New York, 1997, pp. 359-366.

[11] Lester, J. C., Voerman, J. L., Towns, S. G., and Callaway, C. B. Deictic believability: Coordinating gesture, locomotion, and speech in lifelike pedagogical agents. Applied Artificial Intelligence, 1999, 13(4-5), pp. 383-414.

[12] Noma, T., and Badler, N. I. A virtual human presenter. In *Proceedings of the IJCAI Workshop on Animated Interface Agents: Making Them Intelligent*, 1997, pp. 45-51.

[13] Rickel, J., and Johnson, W. L. Integrating pedagogical capabilities in a virtual environment agent. In *Proceedings of the First International Conference on Autonomous Agents*, ACM Press, New York, 1999, pp. 30-38.

[14] Terzopoulos, D., and Waters, K. Physically Based Facial Modelling, Analysis and Animation, Journal of visualization and Computer Animation, 1990, 1(2), pp. 73-90.

[15] Traum, D., and Rickel, J. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, ACM Press, New York, 2002, pp. 766-773.

Virtual Training for Manufacturing and Maintenance based on Web3D Technologies

Demis Corvaglia
HCI Lab, Dept. of Math and Computer Science
University of Udine
via delle Scienze 206, 33100 Udine, Italy
+39 0432 558413
corvagli@dimi.uniud.it

ABSTRACT
In this paper, we (i) introduce the Computer-Based Training (CBT) context, in particular CBT that employs Virtual Reality (VR) technologies; (ii) describe the generic environment we developed to allow the employment of immersive hardware in Web3D applications, (iii) describe the immersive Virtual Training (iVT) application we have built for the training of industrial maintenance personnel.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: *Training, help, and documentation*. I.3.6 [**Methodology and Techniques**]: *interaction techniques.* I.3.7 [**Three-Dimensional Graphics and Realism**]: *virtual reality*.

General Terms
Algorithms, Management, Documentation, Design.

Keywords
Computer-Based Training, Virtual Manufacturing, Virtual Manuals, VRML, X3D, Virtual Reality Environment.

1. INTRODUCTION
Well-designed Computer Based Instruction (CBI) and Training (CBT) application have demonstrated to be effective as learning tools (see, e.g., [15][18][10]). Virtual Reality (VR) is used to improve realism and effectiveness of CBT systems in major fields such as medicine and disabled-care, military and industrial applications. In medicine, VR allows to train surgeons to perform medical operations without risks for patients [16][5]. Moreover, VR training helps the disabled to learn how to deal with every day tasks, avoiding dangerous situations or social discomforts [8]. In military applications, soldiers can virtually try extreme scenarios [24][3] and practice with handling operations in foreign cultures [9]. Since 1991, NASA employs VR [28] where space missions can be tested and astronauts can learn shuttle functionality. Industry, e.g. automotive, is also strongly interested in VR for the training of design, production and maintenance [2] [24]. Sense of Presence (SoP) [20] and interaction richness [17] are primary requisites of a VR training system. To improve these aspects, virtual characters are an effective tool in real world applications where users have to learn hand-operated tasks. Virtual training systems such as Jack [6] and Steve [27] adopted this approach.

In general, VR training offers powerful tools, but it has to be both graphically and physically realistic and comfortable (ergonomic) for trainees.

This paper presents a CBT system for the manufacturing and maintenance contexts based on VR hardware and Web3D technologies. In the considered context, trainees need to interact with mechanical parts, using specific tools (e.g. snapper, screwdriver, …), that can be virtually simulated. VR (especially immersive VR) is here able to: (i) provide an appealing training environment, (ii) avoid dangerous situations, and (iii) reduce training costs [19].

The paper is organized as follows. Section 2 introduces iVRML (immersive VRML), a generic environment we developed to allow the employment of immersive hardware in Web3D applications. We use iVRML to develop the iVT (immersive Virtual Training) system described in Section 3. Section 4 presents an example of iVT applied to jet engine maintenance. Section 5 summarizes the main results of this project, limitations and future work.

2. iVRML (immersive VRML)
VRML (Virtual Reality Modeling Language) [29] is a language for the definition of interactive 3D graphics contents both for standalone applications and for the Web. VRML became an ISO standard in 1997, and the X3D (the language that evolved from VRML) specification [30] has recently been approved by ISO.

iVRML is a generic framework to provide an immersive experience using standard VRML/X3D worlds. iVRML allows to:

- use VR hardware (e.g. HMDs, datagloves, 3DOF/6DOF sensors) in VRML/X3D worlds;
- easily develop specific VR applications, such as the iVT (immersive Virtual Training) described in detail in Section 3.

Section 2.1 briefly introduces the iVRML software architecture and user interface. Section 2.2 discusses VR hardware compatibility and the interface between device drivers and VRML world. Section 2.3 presents an example of a basic iVRML framework extension: user navigation in 3D worlds using hand gestures and head movement.

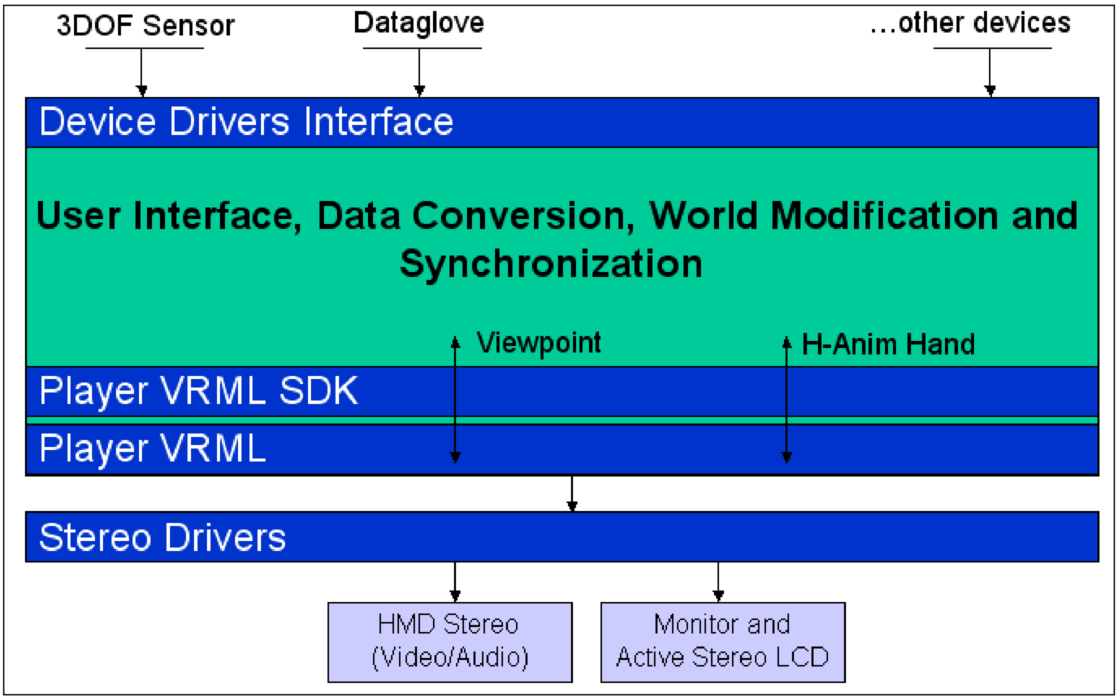


Figure 1. High level iVRML architecture.

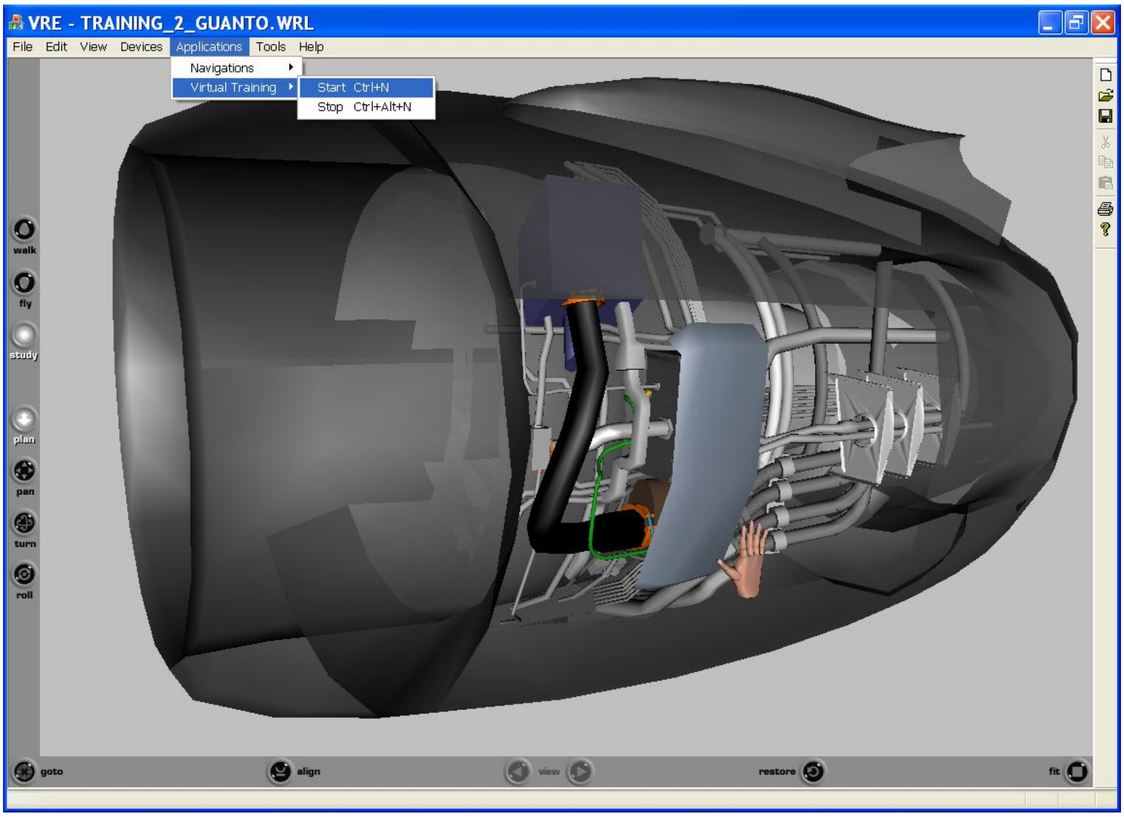


Figure 2. iVRML user interface during a training session on a jet engine (engine model downloaded from [23]).

2.1 ARCHITECTURE AND USER INTERFACE

iVRML architecture is shown in Figure 1. The main container (the biggest box) is a Visual C++ application. It receives data from VR hardware (e.g. 3DOF/6DOF sensors, datagloves); this data is processed by the Device Driver Interface (DDI) that provides methods to access data structures updated by VR devices. First, the application core converts and integrates data from DDI (automatically or under user directives inserted by UI); second it updates the VRML world depending on the application (simulations are computed in reaction to user stimuli). The VRML world is displayed in the main frame of the application, through a Cortona [22] VRML player ActiveX control. VRML scene tree manipulation is performed using Cortona SDK. Common iVRML functions are:

- changing viewpoint orientation according to 3DOF sensor output;
- driving virtual hand according to 6DOF sensor and dataglove (fingers flexure) output;
- detecting and responding to collisions between user controlled objects and other scene objects.

These functions are configurable depending on specific application goals.

The last layers of the diagram shown in Figure 1 concern Stereo Drivers and displays. They allow stereoscopic graphics output. To implement this architecture layer we employed nVidia Stereo Drivers [21] coupled with a GForce graphics board. Stereo video signal is driven to Head Mounted Displays (HMD) for immersive experiences or to CRT monitors (in this case, stereo effect is obtained using active LCD stereo glasses).

Figure 2 shows an iVRML user interface. The application menu allows users to customize basic application behaviour. It concerns:

- opening VRML files;
- activating and setting VR devices;
- connecting VR device data to VRML entities (e. g. 3DOF sensor to a viewpoint);
- enabling special iVRML functionality support, such as object-to-object collision detection, text-to-speech, positional audio and environment sound effects (e.g. reverb).

When user chooses a specific iVRML application (for example, Figure 2 shows “Virtual Training” activation), stored configurations are loaded and iVRML displays dialog windows specifically designed for current task. Applications can extend the specific user interface even in the VRML world using 3D widgets (e.g., 3D buttons or slides, On Screen Displays, …), indeed, VRML events can be connected to application core management.

2.2 VR HARDWARE

So far, we have used iVRML connected to several VR hardware devices, in particular:

- 3DOF sensors (interTrax2 and inertiaCube2 from interSense [13]);
- datagloves (Dataglove 5 from 5DT [1]);
- Head Mounted Displays (HMD) based on the Cy-Visor platform [7].
- Active LCD stereo glasses (Crystal Eyes 3 from StereoGraphics [25]).

iVRML allows one to extend device support easily thanks to DDI (see section 2.1), e.g. to employ 6DOF sensors such as the pciBird from Ascension [4]).

2.3 iVRML NAVIGATION MODES

In 3D virtual worlds, navigation is often the main task. VRML\X3D players allow different navigation modes [30], but all of them concern only user interaction through keyboard, mouse or joystick. Using VR hardware, navigation modes can be improved and tailored to a more immersive experience. In iVRML, we have implemented two major navigation modes:

- **Guided**: user’s viewpoint is tied to an animated scene object (e.g. virtual car); viewpoint orientation is also influenced by 3DOF sensor on HMD.
- **Free**: body position/orientation and velocity are controlled by hand gestures, that are captured by a dataglove (or a joystick); viewpoint orientation is controlled both by body position/orientation and head orientation (this way, the user can move towards a given direction while looking towards a different one). This mode is implemented by combining body position/orientation and head orientation.

Figure 3 and Figure 4 show the two navigation modes. In Figure 3, the user is in Guided mode on board of an aircraft and he is experiencing the manouvres of an aerobatic team [11]. In Figure 4, the user is in Free navigation mode, exploring a cultural heritage reconstruction [26].

Figure 3. Guided navigation mode. User position is constrained (e.g. inside an aircraft); Viewpoint orientation changes following user’s head (3DOF sensor mounted on HMD).

Figure 4. Free navigation mode. Body position, velocity and orientation are driven by hand gestures; viewpoint follows user head orientation (as in the real world).

3. The iVT (immersive VIRTUAL TRAINING) application

IVT (immersive Virtual Training) is a CBT system implemented over iVRML. Industrial maintenance procedures are often very well defined and tasks sequence can be very elaborate. The learning process is effective if users can directly work on physical components. iVT provides this opportunity without risks (both for the maintainer and for the components). iVT assists the trainee during training sessions; it suggests the correct task to perform through a textual description (optionally converted into audio by a text-to-speech engine). Current iVT implementation supports right hand interaction based on a dataglove and a tracking sensor mounted on it. When iVT suggests the task to perform, the user can move her virtual hand and execute the task. Collision detection between virtual hand and scene objects is computed in real-time. The hand structure follows the H-Anim specification [12]; it is provided with spherical collision detection sensors (see Figure 5) that are used to determine hand-object proximity or contact.

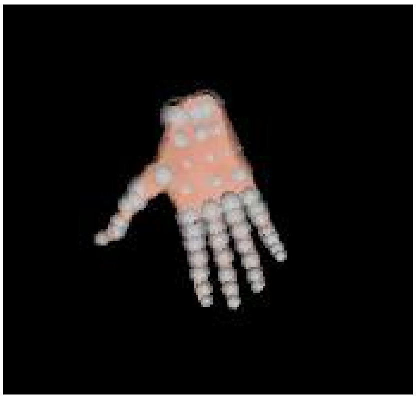


Figure 5. Spherical sensors of the virtual hand.

iVT offers low level support to develop virtual training application. iVT administrator (i.e. the person who defines the tutorial) instantiates and extends a language whose syntax is standard VRML and whose structures is shown in Figure 6.

We have designed specific PROTOs to describe:

- ***Mechanical Items***: bolts, nuts, screws, …;
- ***Tools***: snappers, pullers, screwdrivers, …;
- ***Training Tasks***: a sequence of basic operations;
- ***Basic Operations***: user's actions and associated conditions (temporal and spatial) checked by iVT.

An example of a task with some basic operations is the following: the trainee has to remove a metal cover from a jet engine (see Figure 7), she has to move her virtual hand towards the item (basic operation 1) and grab it (basic operation 2). iVT tests the correct grab conditions and when these are satisfied, a visual effect advises user. After this, hand and mechanical item have to move together, both controlled by the dataglove. When the user reaches the required release position (basic operation 3), the item disappears and the system indicates the new task to perform.

Every mechanical item has its specific assembling/disassembling conditions and associated mechanical tool. iVT is developed for common mechanical tasks, but the administrator can define new ones. For example, if a mechanical item needs a special tool, the administrator has to define item and tool geometry, start position of the item inside the complete mechanic part, insertion and extraction conditions for the item, tool and item interaction rules. This definition can involve complex interaction that requires to write ad-hoc javascript code. However, for learning purposes, it is sufficient to define item-tool(s) association, point of interaction and the tool-hand(s) grabbing conditions. When a correct tool-item (relative) position is reached, tool usage can be demonstrated by 3D animation.

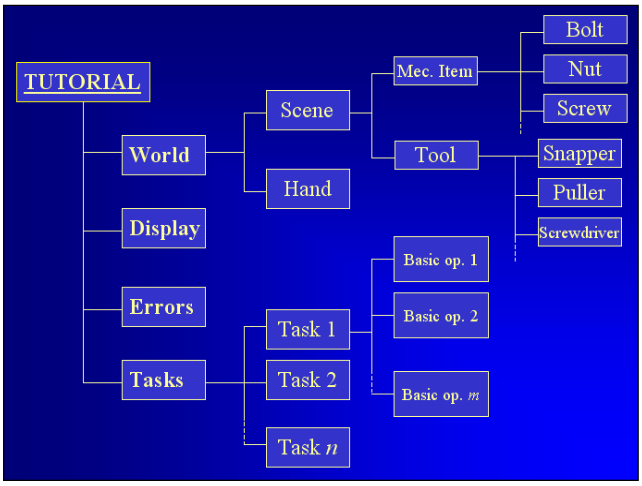


Figure 6. Language abstract diagram designed to define a tutorial in iVT.

A (human) trainer (that can be the administrator or a high level supervisor) can define messages that will be displayed during the training, and for error conditions and warnings.

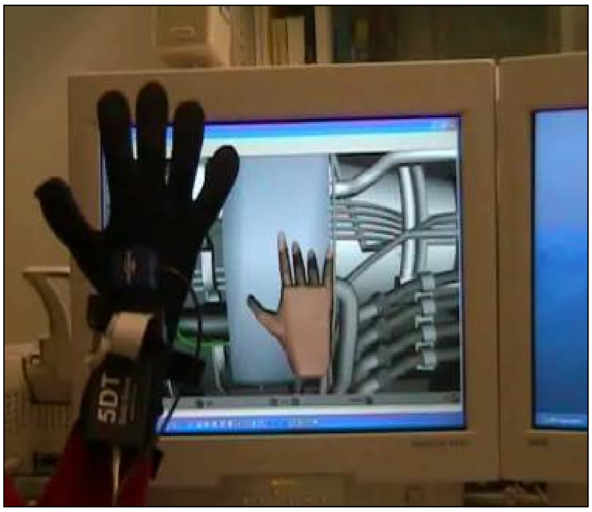


Figure 7. A iVT user is learning how to remove a cover from the jet engine.

4. iVT CASE STUDY: DISASSEMBLING AN AIRCRAFT JET ENGINE

We have experimented iVT on the 3D model and disassembly procedures available at [23]. While the desktop version of the virtual manual is limited to showing the tasks to the user, our re-implementation in iVT allows the user to virtually try the disassembling phase with her hand. Figure 8 and Figure 9 show two training examples. The first requires to unscrew a bolt to free a cover; iVT uses an On Screen Display (OSD) to tell the user how to perform this task. This task is followed by cover removing shown in Figure 7. In Figure 9, the user removes a tube; this task requires correct hand posture.

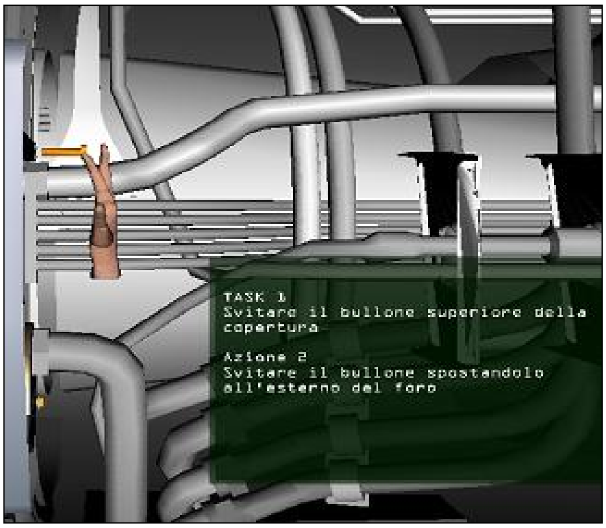


Figure 8. Unscrew the bolt (using "finger-screwdriver").

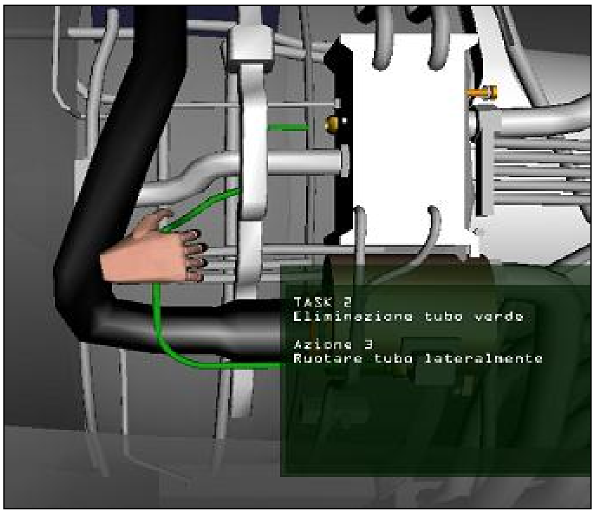


Figure 9. Green tube removal. The OSD tells to user: "Turn tube sideways".

5. CONCLUSIONS

In this paper, we presented mainly our iVRML and iVT applications. Informal user testing sessions produced encouraging results about iVT potential. Users greatly appreciated the possibility of virtually trying in a direct way the actions that are usually only displayed in virtual manuals. We plan to perform controlled experiments when more iVT functionalities will be added. Indeed, there is a number of limitations of the system on which we are working:

- enabling two-handed interaction;
- enabling force-feedback and haptic devices support;
- implementing object-to-object collision detection;
- designing and implementing a visual user interface for authoring training materials.

An important future development will consist in the introduction of virtual characters that demonstrate correct tasks execution to users.

6. ACKNOWLEDGMENTS

Elisa Cazzitti played an important role in the implementation and testing of the first version of the jet engine application.

7. REFERENCES

[1] 5DT home page: www.5DT.com.

[2] A. Gomes de Sà, G. Zachmann, *Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes. Computer & Graphics*, Elsevier, Vol. 23, No. 3, pp. 389-403, 1999.

[3] A. W. Stedmon, *Re-viewing Reality: Human Factors of Synthetic Training Environments*. International Journal of Human-Computer Studies, No. 55, pp. 675-698, 2001.

[4] Ascension Technology home page: http://www.ascension-tech.com.

[5] B. Fröhlich, G. Grunst, W. Krüger and G. Wesche, *The Responsive Workbench: a Virtual Environment for*

Phisycians. Comput. Biol. Med., Vol. 25, No. 2, pp. 301-308, 1995.

[6] Badler N. I., Philips C. B., Webber B. L., *Simulating Humans, Computer Graphics Animation and Control*, Oxford University Press, New York 1993.

[7] Cy-Visor HMD: http://www.personaldisplay.com.

[8] D. Gourlay, K.C. Lun, Y. N. Lee and J. Tay, *Virtual Reality for Relearning Daily Living Skills*. International Journal of Medical Informatics, No. 60, pp. 255-261, 2000.

[9] E. Sims, *Using Extensible 3D (X3D) to Repurpose Modeling and Simulation Assets for Distributed Learning.* Technical Report, www2.vcom3d.com/SISOSpring03.pdf.

[10] F. Mantovani, *VR learning: Potential and Challenges for the Use of 3D Environments in Education and Training*, Chapter 12 in Towards Cyber-Psychology: Mind, Cognitions and Society in the Internet Age, Amsterdam, IOS Press, 2001.

[11] Frecce3D web site: www.frecce3D.uniud.it.

[12] H-Anim (Humanoid Animation Working Group) home page: www.hanim.org.

[13] InterSense home page: www.intersense.com.

[14] J. R. Wilson, *Virtual Environments Applications and Applied Ergonomics*. Applied Ergonomics, No. 30, pp. 3-9, 1999.

[15] J.D. Fletcher, *Does the Stuff Work? Some Findings from Applications of Technology to Educational and Training*. Proc. of Conference on Teacher Education and the Use of Technology Based Learning Systems, 1996.

[16] K. Brodlie, N. El-Khalili and Y. Li, *Using Web-based Computer Graphics to Teach Surgery*. Computer & Graphics, No. 24, pp. 157-161, 2000.

[17] L.Sastry, D.R.S. Boyd, *Virtual Environments for engineering applications*, Virtual Reality 3 (1998) p.235-244.

[18] M. Barnes, *Virtual Reality and Simulation*. Proc. of the 28th Conference on Winter Simulation, Colorado, California, pp.101-110, 1996.

[19] M. Ponder, B. Herbelin, T. Molet, S. Schertenlieb, B. Ulicny, G. Papagiannakis, N. Magnenat-Thalmann, D. Thalmann, *Immersive VR decision training: telling interactive stories featuring advanced virtual human simulation technologies*. 9th Eurographics Workshop on Virtual Environments IPT/EGVE2003, pp. 97-106, Zurich, Switzerland, 2003.

[20] M.Slater, *Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire*, Presence 8 (1999) p.73-79.

[21] NVidia home page: www.nVidia.com.

[22] Parallel Graphics home page: http://www.parallelgraphics.com.

[23] Parallel Graphics, Virtual Manuals Gallery, http://www.parallelgraphics.com/virtual-manuals/gallery/engine2.

[24] R. Stone, *Virtual Reality for Interactive Training: an Industrial Practitioner's Viewpoint*. International Journal of Human-Computer Studies, No. 55, pp. 699-711, 2001.

[25] StereoGraphics home page: www.stereographics.com.

[26] Udine3D web site: www.udine3D.uniud.it.

[27] W. L. Johnson, and J. Rickel, *"Steve: An Animated Pedagogical Agent for Procedural Training in Virtual Environments"*, Sigart Bulletin, ACM Press, vol. 8, number 1-4, 16-21, 1997.

[28] W. R. Sherman, *Experiences with Virtual Reality Applications.* Proc. 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 473-476, 1997.

[29] Web3D Consortium home page: www.web3D.org.

[30] X3D specifications: http://www.web3d.org/x3d/specifications/index.html.

WebTOP: A 3D Interactive System for Teaching and Learning Optics

Matthew Morris
Department of Computer Science and Engineering
Mississippi State University
Mississippi State, MS 39762
mjm9@msstate.edu

John T. Foley
Department of Physics and Astronomy
Mississippi State University
Mississippi State, MS 39762
jtf1@msstate.edu

Taha Mzoughi
Department of Physics and Astronomy
Mississippi State University
Mississippi State, MS 39762
tm32@msstate.edu

ABSTRACT

WebTOP is a three-dimensional, web-based, interactive computer graphics system that helps instructors teach and students learn about waves and optics. Current subject areas include waves, geometrical optics, reflection and refraction, polarization, interference, diffraction, lasers, and scattering. These modules offer many features which lend themselves to classroom use or self-guided study. It is implemented using VRML, Java, and VRML's Java EAI.

Categories and Subject Descriptors

J.2 [**Physical Sciences and Engineering**]: Physics

Keywords

WebTOP, VRML, EAI, Java, XML, optics, waves, reflection, refraction, geometrical optics, polarization, interference, diffraction, scattering, lasers.

1. INTRODUCTION

The Optics Project on the Web (WebTOP) is a web-based 3D interactive computer graphics system that simulates optical and wave phenomena [2, 3, 4]. Its purpose is to help students and teachers better understand these phenomena. It is designed to be flexible, allowing faculty and students to use it in a variety of ways. Currently, WebTOP includes sixteen different modules: Waves, Lenses, The Eye, Reflection and Refraction - Vectorial, Reflection and Refraction - Waves Two Media, Reflection and Refraction - Waves Three Media, Polarization, Michelson Interferometer, Fabry-Perot Etalon, Fraunhofer N-Slit, Transmission Grating, Rayleigh Resolution, Fresnel Single Slit, Fresnel Circular, Scattering, and Lasers.

All of the modules share certain basic features. Each includes the simulation, an overview of the theory behind the topic, a set of example sessions that can be replayed, and a set of suggested exercises. In addition, WebTOP-based, guided tutorials on various topics are being developed. These tutorials will take students through the material step by step, providing visual examples and inquiry-type exercises designed to enhance the theory beyond the capability of a traditional textbook.

WebTOP is implemented using VRML, Java, VRML's Java EAI, and JavaScript. The project currently runs under Internet Explorer with the Blaxxun Contact plugin and is available at http://webtop.msstate.edu.

$

2. RELATED WORK

There are numerous applications where VRML is used for teaching science. The VRML Gallery of Electromagnetism [5], for example, is an exhaustive showcase of more than 40 visualizations of electromagnetic phenomenon. While these "static" models clearly provide insight into this subject, they do not allow user interaction beyond the usual VRML exploration methods.

There are other exemplary web-based 3D simulations in the physical sciences [7, 8]. Many employ some simple animation and interaction using the standard VRML constructs, but they do not offer fine grained user interaction. WebTOP's approach is similar, but uses a combination of VRML, Java EAI, and Java Script to allow the user control over all varibles in the simulation, thereby leading to a more thourough exploration of each concept.

3. OVERVIEW OF A MODULE

3.1 3D Scene

A WebTOP simulation consists of three parts (see Fig. 1). First is the Scene in which the simulation is presented. The primary controls are the navagation icons which provide easy navigation through the three-dimensional space. The arrows icon allows the user to zoom into and out of a scene. The small globe allows for three-dimensional rotation, and the crossed arrows icon allows the user to pan the scene around a two-dimensional plane. There are also widgets that allow for the user to change the parameters of the simulation. For example, in Figure 1 there is a wheel widget which can be spun to control the wavelength of the light. Likewise, the positions of many scene elements can be changed using a widget. Observation screens, which are used in many modules, can be moved by dragging their red double cone widget.

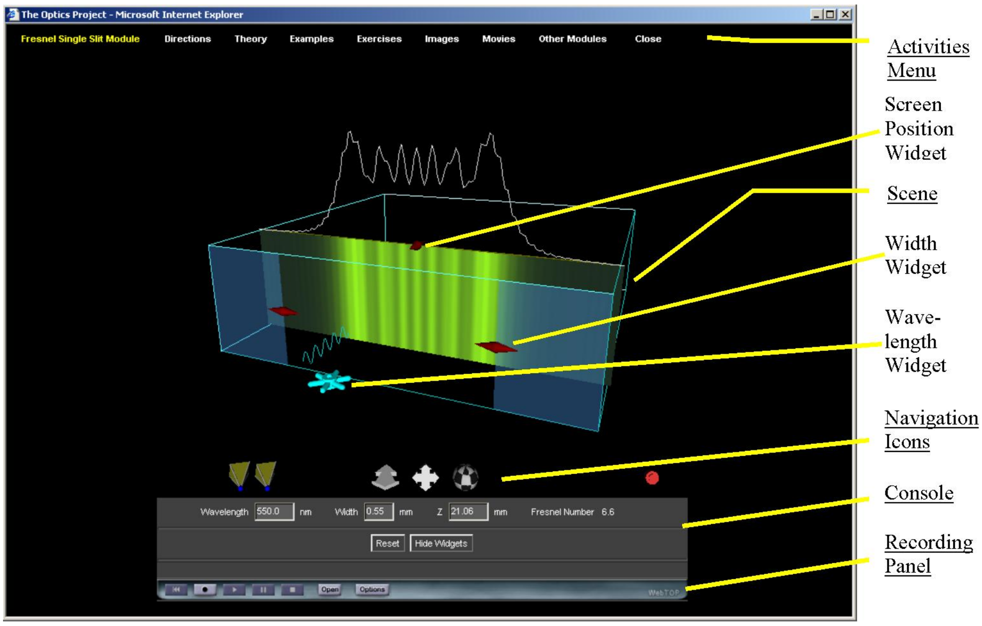


Figure 1: The Fresnel Single Slit Module. 1:Activities Menu 2:Screen Position Widget 3:Width Widget 4:Wavelength Widget 5:Navagation Icons 6:Console 7:Recording Panel

3.2 Console

Another important part of a module is the Java Console which allows for user input to the simulation. It often contains number boxes that control numerical parameters such as wavelength, aperture size, or angle of incidence. This panel may also contain other controls such as drop down windows which allow for a choice from a list of options. These choices may be the type of lens: concave, convex, etc.., or it may allow the user to control the image quality: very high, high, ..., lowest. The console allows the user to control the parameters of the simulation in a more precise manner than the widgets.

3.3 Recording Panel

The Recording Panel is at the bottom of the WebTOP window. It contains VCR-like controls that allow the user to record a WebTOP session as an XML-formatted script, and then play the script back at a later time. The scripts are recorded as .wsl files (WebTOP scripting language files). They record (a) each action performed, e.g., a parameter or viewpoint change, and (b) the time that has elapsed since the last action was performed. These files are easy to edit, and this allows the user to adjust the parameter values, the viewpoints, and the rate at which actions are performed during the playback.

The ability to record a script for later playback is useful in a number of situations. For example, for an in-class presentation that involves a series of events, a professor can record his session as a script before class, and then simply play the script in class, instead of having to perform all the actions in class in real-time. Homework is another area of use. Students can be assigned problems in which they must interact with a module in order to gain understanding of the underlying physics. These interactions can be saved as a script and emailed to the professor for grading.

4. THE MODULES

4.1 The Waves Module

The Waves module (see Fig. 2) is a simulation of waves in a ripple tank. One or more monochromatic point sources and/or line sources can be placed on the surface of the water. The resulting disturbance is shown either as a still picture or as traveling waves. A widget is associated with each point source, allowing the user to interactively change the source position, amplitude, wavelength, and initial phase of the generated wave. A similar widget allows the user to change the parameters of each line source.

4.2 The Lenses Module

This module simulates the behavior of light rays as they pass through a system of lenses and stops on an optical bench (see Fig. 3). The user can choose from amongst several different objects: an on-axis point source, an off-axis point source, a point source at infinity, five point sources in the shape of a T, etc. The user can put an unlimited number of lenses and stops on the bench. The position, diameter and focal length of each lens can be varied interactively, as can the position and diameter of each stop. Each point source on the object emits a large number of rays in random directions, and these rays travel through the system according to the laws of paraxial geometrical optics. A movable observation screen allows the user to see the ray distribution in any plane perpendicular to the axis of the bench.

4.3 The Eye Module

The Eye module simulates the formation of images in the human eye. The eye looks at an object that can be moved, and the resultant change of the image on the retina is simulated. This module can also be used to show the effects of hyperopia, presbyopia, and myopia. The consequences of these conditions can then be corrected by the addition of an

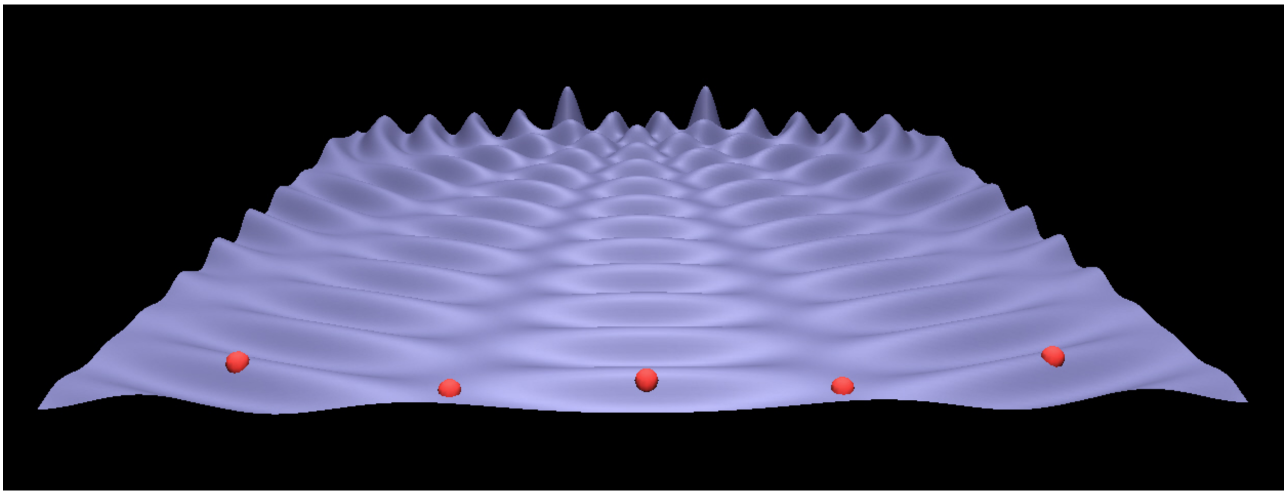


Figure 2: The Waves module. Two point sources (in the background) create a series of maxima and minima on the surface of the ripple tank. The first, third, and fifth floating red balls are at locations where constructive interference is occurring. The second and fourth balls are at places where destructive interference is occurring.

eyeglass or contact lens.

4.4 Reflection and Refraction - Vectorial

This module simulates the incident, reflected, and refracted electric field vectors generated when a monochromatic plane wave is incident upon a planar interface that separates two media with different indices of refraction, or when unpolarized incident light is used (see Fig. 4). The user controls the characteristics of the incident light and the indices of refraction of the two media. This allows several important phenomena to be simulated, including total internal reflection, polarization upon reflection, and the various cases of phase shifts upon reflection.

4.5 Reflection and Refraction - Waves Two Media Module

This module displays a monochromatic plane wave of s-polarized light incident upon a planar interface separating two media with different indices of refraction. In contrast to the Vectorial module, this module uses a ripple tank to show the wave functions of the fields, instead of the electric field vectors. The user can choose to display the incident wave, the reflected wave, or the superposition of the two in the first medium. Controlleable parameters are the wavelength, amplitude, and the angle of incidence of the incident light, and the indices of refraction of the two media.

4.6 Reflection and Refraction - Waves Three-Media Module

This is an extension of the Waves Two Media module to include three media whose indices of refraction can be varied. This module allows the user to see what happens during phenomena such as frustrated total internal reflection (see Fig. 5) and thin film interference.

4.7 Polarization

The Polarization module simulates the propagation of the electric field vectors of polarized or unpolarized light, and the effects of various optical elements, e.g., linear polarizers and wave plates, on the state of polarization of the light (see Fig. 6). The user can add as many optical elements as is desired, and controls the properties of the incident field and the characteristics of each optical element.

4.8 Michelson Interferometer

This module simulates light from a monochromatic point source incident upon a Michelson interferometer (see Fig. 7). The resulting intensity pattern is displayed on an observation screen, and intensity as a function of position is shown as a graph above the observation screen. The parameters that can be varied are the wavelength of incident light, the rotation angle of the tilt mirror, and the position of the translation mirror.

4.9 Fabry-Perot Etalon

In this module, light from a monochromatic point source is incident upon a dielectric slab whose surfaces are coated with a reflective coating. The resultant transmitted intensity is an interference pattern that is displayed on an observation screen; a graph of the intensity is also provided. The user controls the wavelength of the incident light, the reflectivity of the slab, and the thickness and index of refraction of the slab.

4.10 Fraunhofer N-Slit

The Fraunhofer N-Slit module simulates a plane wave of monochromatic light that is normally incident upon a plane that contains N slits. The intensity pattern is displayed on an observation screen, and, as in previous modules, a graph of the intensity as a function of position is shown above the observation screen. The user has control over the wavelength, number of slits, slit width, distance between slits, and position of the observation screen.

4.11 Transmission Grating

This module simulates light from a discharge tube incident upon a diffraction grating. The resultant intensity pattern is displayed on a semicircular observation screen. The user can choose from amongst several standard dischage tubes (hydrogen, helium, mercury, ...) or create a source with a

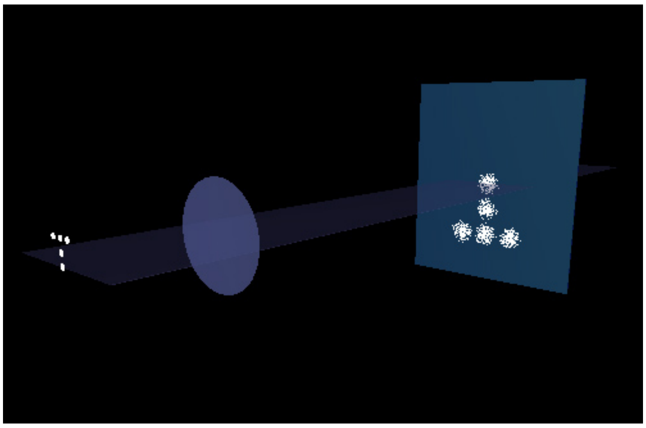


Figure 3: The Lenses module. The location of the observation plane does not coincide with the image plane.

spectrum he provides. The user also controls the number, width, and separation of the grooves of the grating.

4.12 Rayleigh Resolution

In this module monochromatic light from two distant point sources separated by a small angle is incident upon a lens (see Fig. 8). The resulting intensity pattern is viewed on an observation screen positioned in the focal plane of the lens, and a graph of the intensity as a function of position across the center of the pattern is displayed above the observation screen. The user can vary the wavelength of the light, the angle between the sources, and the diameter of the lens. Each stellar image is, in fact, a small diffraction pattern, and by varing the parameters, the user learns how they affect whether or not the two images can be resolved.

4.13 Fresnel Single Slit

In this module a monochromatic plane wave of light is incident upon a single slit and the diffraction pattern close to the slit, i.e. in the Fresnel region, is investigated. Variable parameters include the wavelength of the incident light, the slit width, and the position of observation screen. The parameters have ranges imposed on them which allow the user to explore the Fresnel region and the beginning of the Fraunhofer region.

4.14 Fresnel Circular

This module is similar to the Fresnel single slit configuration, however it uses a circular aperture or obstacle. In the latter case the famous "spot of Arago" is displayed on the optical axis.

4.15 Scattering

This module simulates the non-resnant scattering of light from an atom. The atom is located at the origin of the scene, and incident light travels along the y-axis. An oscillating vector shows the induced dipole moment of the atom. The scattered electric field vectors are depicted at sets of observation points along three different axes: x, z, and a moveable axis whose angular position is controlled by the user. The incident light can be either linearly polarized or unpolarized.

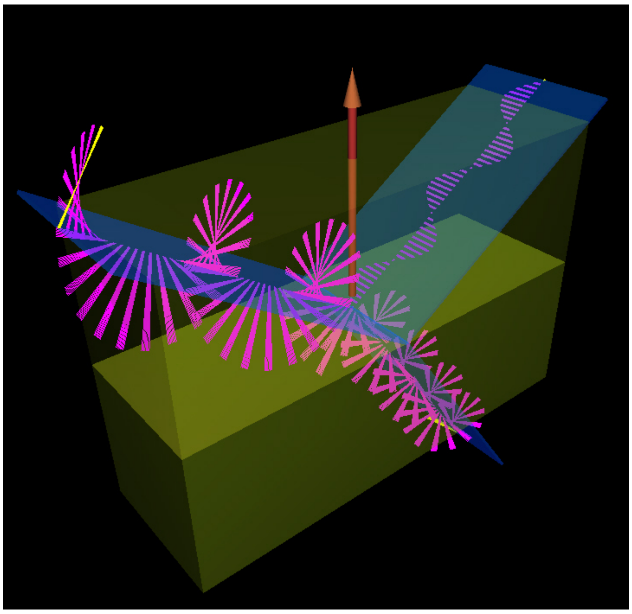


Figure 4: The incident, reflected, and transmitted electric field vectors when circularly polarized light is incident (from the left) from air onto glass at the polarization angle (Brewster's angle). The reflected light is linearly polarized. The brown arrow denotes the normal to the planar interface.

4.16 Lasers

This module simulates a laser beam from an laser cavity oscillating in one of the following transverse modes: $\mathtt{TEM}_{00}$, $\mathtt{TEM}_{10}$, $\mathtt{TEM}_{01}$, or $\mathtt{TEM}_{11}$ (see Fig. 9). The cavity has a totally reflecting spherical mirror on one end, and a partially reflecting mirror on the other. The intensity pattern of the light emitted by this laser is displayed on an observation screen. The user can vary the wavelength of the light, the radius of curvature of each mirror, the length of the cavity, and the position of the observation screen.

4.17 Implementation

WebTOP is a highly organized software system. Software components are modular and duplicate code is minimized. This is accomplished through the use of libraries which manage Java-VRML interaction, widget creation, animation, scripting, common look-and-feel, and common mathematical functions.

For example, the use of a common set of widgets across modules decreases the slope of the learning curve for new users, and at the same time makes maintance easier. These widgets are kept in a single file, Widgets.wrl, and each module that needs to use one can do so through the use of VRML's EXTERNPROTO. Similarly, there are a number of shared resources available in the util library. The AnimationEngine, for instance, is used in five different modules.

It is a thread-safe general purpose animation controller which provides the programmer clean access to Play, Pause, and Step constructs, as well as control of the animation's period.

Another important implementation component is the WApplet base class. This class implements features common to

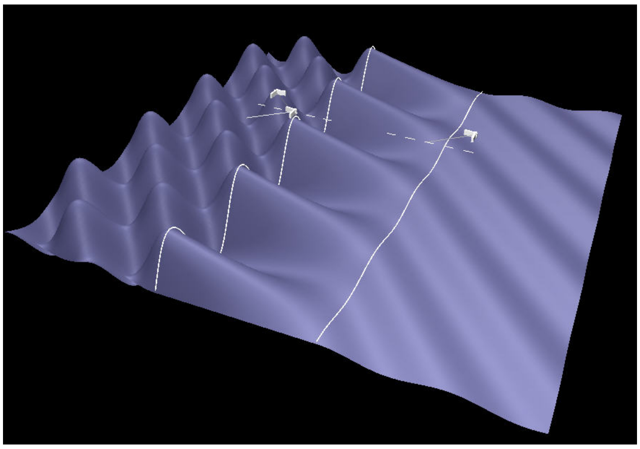


Figure 5: The Reflection and Refraction - Waves Three Media module. Frustrated Total internal reflection is depicted.

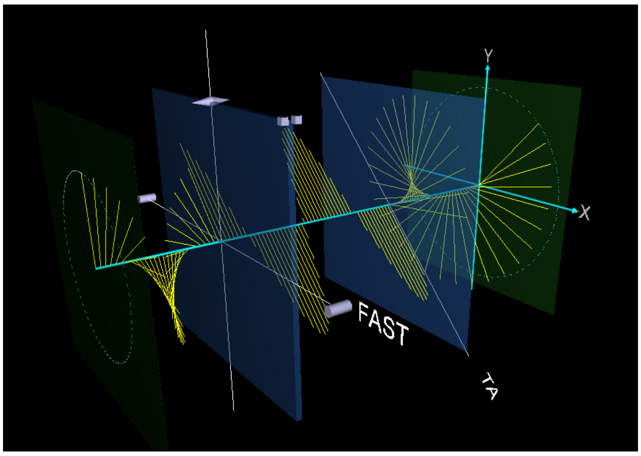


Figure 6: The Polarization module. Right circularly polarized light, traveling from right to left, is incident upon a linear polarizer. The resultant linearly polarized light is then incident upon a quarter-wave plate, creating left circularly polarized light.

all modules. This includes setting up the Navigation and WSL Panels; managing the scripting of widgets, number boxes, and other user input; and providing commonly used EAI functionality. The WApplet class is a relatively new addition to WebTOP, but it has already proven able to drasically reduce the size and complexity of a given module. The Fabry Perot module, for example, is now only 260 lines of Java code. All the programmer must do to begin implementation of a new module is define the number boxes and widgets, connect them together using WebTOP coupler and scripter objects, fill in `setupVRML()` and `setupGUI()` methods, and define any methods necessary to perform module-specific math.

5. USING WEBTOP FOR TEACHING

WebTOP is designed to be used in class and outside of class. It is currently being used in more than twenty-five universities in introductory physics classes (both algebra and calculus based) and in upper-level undergraduate optics courses.

WebTOP can be used for in-class demonstrations. Its three-dimensional nature allows the presenter to show the phenomenon from the most advantageous viewpoint. The ability to change the values of all the relevant parameters in a simulation allows the instructor to show, in real time, the effect of changing each parameter in a way that can be seen clearly by the students, even in large lecture halls. WebTOP can be used in addition to, or instead of, regular equipment-based demonstrations.

More importantly, using WebTOP for in-class presentations lends itself to Interactive Lecture Demonstration type activities, providing an ideal mechanism for creating an active classroom learning environment [6]. Students are asked to predict the outcome of the demonstration and to justify their answers. Following the ensuing discussion, the teacher can then demonstrate the phenomenon. In cases when most of the students provide wrong predictions, the teacher can ask them to help justify the correct answer.

Furthermore, WebTOP can be used for homework. Students can use WebTOP to help verify the answers they get in regular end of the chapter problems. Additionally, teachers can assign some of the suggested WebTOP-based problems provided in the "Exercises" section of the modules. In this case, the students can be asked to analyze certain situations, produce numerical or qualitative answers to questions, and then use WebTOP to simulate the situation and check their answers. The students can be asked to turn in their written answers to the questions, relevant screen captures from WebTOP modules, or WebTOP scripts. Homework can also take the form of an online tutorial. Students can be asked to learn about a particular topic by completing an online WebTOP guided tutorial. The tutorial can be assigned before or after lecture instruction.

WebTOP can as well be used to supplement laboratory activities. It can be used in pre-lab activities to help explain the phenomenon that is going to be investigated in the lab. It can be used during the lab to compare actual data to the simulation results and to help draw inferences when equipment limitations occur.

Finally, WebTOP can be used for student projects. Students can work in teams or individually on particular problems and use WebTOP to help illustrate their presentations and reports.

At Mississippi State University WebTOP has been used and evaluated by students in five sections of our calculus-based introductory physics course and two sections of our algebra-based introductory course over the last three years. The seven sections had a total of 335 students.

WebTOP was used in these classes in the two ways. First, it was used to simulate wave and optical phenomena in class. Secondly, the students were required to do two homework sets in which they needed to use WebTOP. In the homework problems they were asked to analyze certain situations, produce numerical or qualitative answers to each question, and then use WebTOP to simulate the situation and check their answers. The students then turned in their written solutions and a screen shot of the WebTOP simulation of each problem.

In the student evaluation of WebTOP performed at the end of each semester, the students were asked, "How useful did you find the WebTOP demonstrations during class

Class - Question 1	Mean	Std. Dev.	Students
Cal Based Group 1	4.56	0.69	43
Cal Based Group 2	4.53	0.57	55
Cal Based Group 3	4.51	0.54	47
Alg Based Group 1	4.54	0.63	52
Alg Based Group 2	4.51	0.64	53
Cal Based Group 4	4.44	0.60	34
Cal Based Group 5	4.57	0.60	51

Table 1: Student responses to Question 1 of the WebTOP evaluation questionnaire (Scale: 5 = very useful, 4 = pretty useful, 3 = somewhat useful, 2 = hardly useful, 1 = not at all useful.)

Class - Question 2	Mean	Std. Dev.	Students
Cal Based Group 1	4.37	0.72	43
Cal Based Group 2	4.27	0.72	55
Cal Based Group 3	4.17	0.91	47
Alg Based Group 1	4.17	0.99	52
Alg Based Group 2	4.33	0.73	53
Cal Based Group 4	4.29	0.71	34
Cal Based Group 5	4.33	0.70	51

Table 2: Student responses to Question 2 of the WebTOP evaluation questionnaire (Scale: 5 = very useful, 4 = pretty useful, 3 = somewhat useful, 2 = hardly useful, 1 = not at all useful.)

for visualizing and understanding the optical phenomena in this course?" The classes were very consistent in responding that they found the in-class demonstrations "very useful" (see Table 1). The students were also asked, "How useful did you find the WebTOP homework for visualizing and understanding the optical phenomena in this course?" In this case they rated WebTOP somewhere in between "very useful" and "pretty useful" (see Table 2).

WebTOP has also been used and evaluated twice in our junior/senior level optics course. It was used for in-class presentation, for homework by the students, and for class projects. The questions in the evaluation questionnaire were long-answer, not numerical. The responses of the students were very positive; each student both semesters answered affirmatively to the question "Did you find the software useful for visualizing and understanding the optical phenomena considered in this course?."

6. CONCLUSIONS

WebTOP uses web3D technology to simulate optical phenomenon. It is a tool that students and educators can use to better understand the principles of waves and optics. The human visual system has the greatest bandwidth to the brain [1], and utilizing its capacity to interpret position, spatial variance, and color is an excellent way to communcate complicated concepts.

7. ACKNOWLEDGMENTS

WebTOP is supported by the National Science Foundation Grant DUE 0231217. We would also like to acknowledge the contributions made to WebTOP by Dr. David C. Banks, Kiril Vidimce, and Yong-Tze Chi.

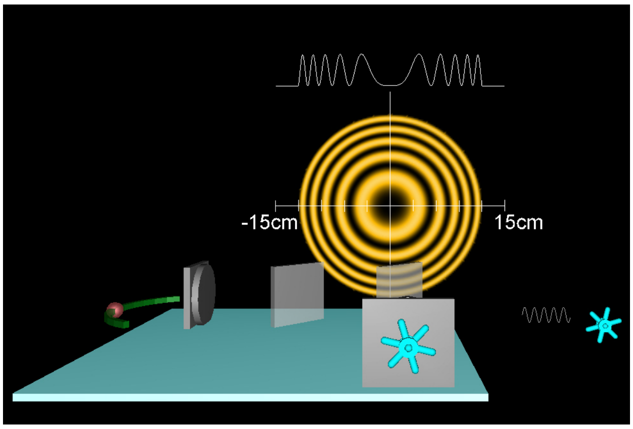


Figure 7: A Michelson interferometer with monochromatic light incident from the right.

8. ADDITIONAL AUTHORS

S. Davis Herring(Department of Applied Science, University of California at Davis, email: `herring@lanl.gov`).

9. REFERENCES

[1] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[2] J. T. Foley, D. C. Banks, Y.-T. Chi, and T. Mzoughi. Webtop: Interactive optics on the web. In *Proceedings of the Web3D-VRML 2000 Fifth Symposium on Virtual Reality Modeling Language*, pages 125–129. Web3D, 2000.

[3] J. T. Foley, K. Vidimce, D. C. Banks, and T. Mzoughi. Webtop: Interactive 3d optics simulations on the web. In *Education and Training in Optics and Photonics*, pages 214–222. SPIE, 2000.

[4] T. Mzoughi, J. T. Foley, S. D. Herring, M. Morris, and B. Wyser. Webtop: Web-based interactive 3d optics and waves simulations. *International Journal of Continuing Engineering Education and Life-Long Learning, in press.*

[5] R. Salgado. Virtual gallery of electromagnetism. http://physics.syr.edu/courses/vrml/electromagnetism/.

[6] D. Sokoloff and R. Thornton. Using interactive lecture demonstrations to create an active learning environment. *The Physics Teacher*, 35:340, 1997.

[7] D. T. University. Virtual reality modeling lanugage in chemistry. http://www.pc.chemie.tu-darmstadt.de/research/vrml/index.html.

[8] VAKHUM. Virtual animation of the kinematics of the human for industrial, educational and research purposes. http://www.ulb.ac.be/project/vakhum/vrml/vrml.htm.

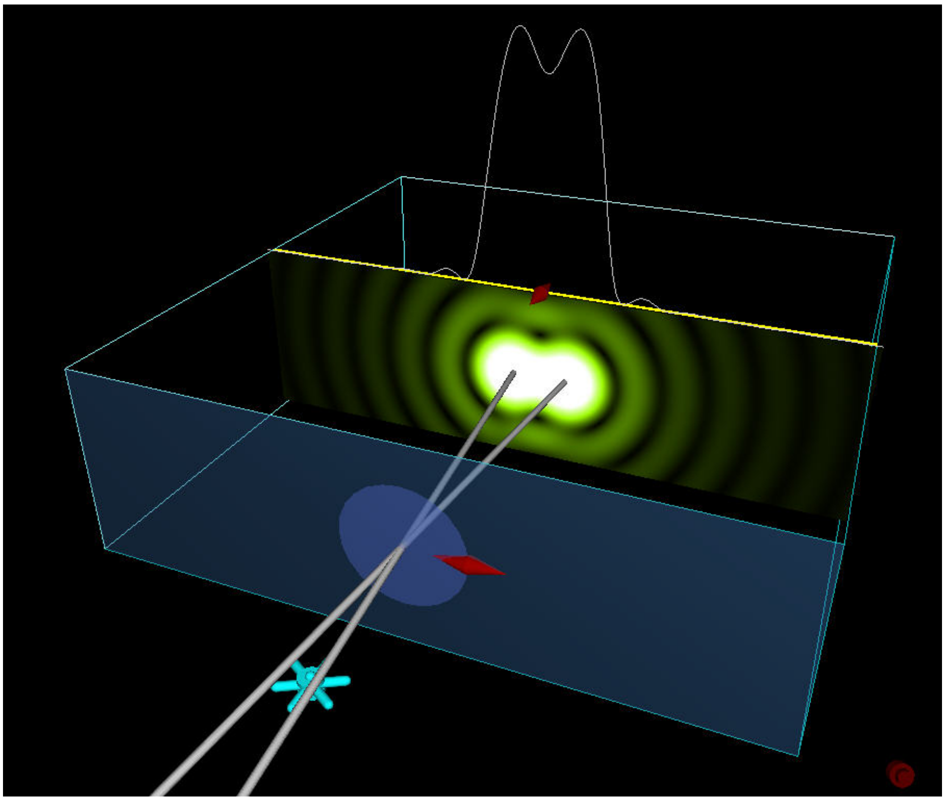


Figure 8: The Rayleigh Resolution module. The angular separation of the two distant sources and the wavelength of the light is such that the two images are said to be "barely resolved."

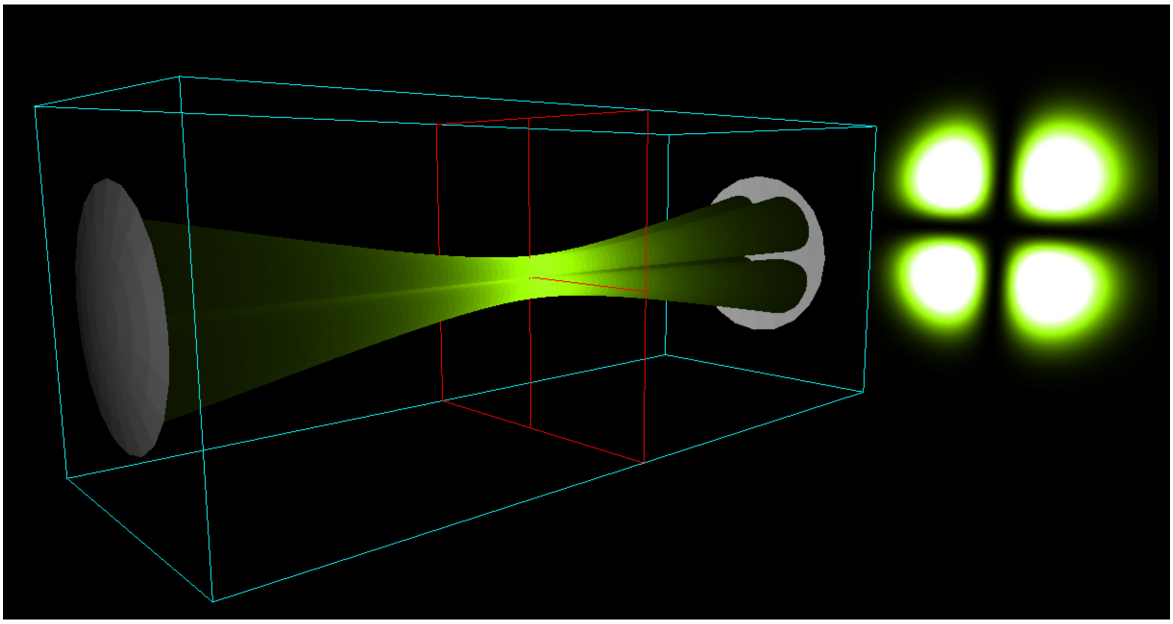


Figure 9: The Laser module with the laser oscillation in its TEM 11 mode.

A Java3D-based Tool for the Development of Virtual Robotic Environments

Eleonora Fantini, Monica Reggiani, Stefano Caselli
Dipartimento di Ingegneria dell'Informazione, Università di Parma
Parco Area delle Scienze, 181A
Parma, Italy
{efantini, reggiani, caselli}@ce.unipr.it

ABSTRACT

The continuously emerging of new markets in robotics needs to be supported by a corresponding improvement in education and training. As advanced topics in robotic courses require the use of expensive equipment, currently students' experience with real robots is often limited thus introducing difficulties in their learning process. This paper describes the ongoing development of an educational tool whose final goal is to simplify the creation of virtual environments allowing users to visualize and interact with three-dimensional robots. The system is implemented using open technologies (Java, Java3D, XML) to provide full portability and interoperability on several computer platforms. Its usefulness has been proven also in research for the implementation of a graphical interface for motion planning applications and for trajectory representations.

1. INTRODUCTION

In the latest ten years, the worldwide market for robotics has been continuously growing. While car industry is still the leading application area, new markets are emerging such as service robotics. Several projects have been undertaken and are leading to credible prototypes and implementations as well as to commercial products in applications ranging from impaired people assistance to automation of gas stations [5]. This expansion in the number of development sectors needs to be supported by a corresponding improvement in education and training.

The availability of cheap robot equipment for mobile robotics has already promoted the growth in the use of robots in undergraduate studies [15], yet the simple hardware does not allow students to get in touch with more advanced topics, such as understanding manipulator mechanics. Students get often stuck with a common early exercise in robotics course, i.e. learning the procedure for assigning the Denavit-Hartenberg parameters [4], the de-facto standard to describe a manipulator configuration. Only when the parameters have been assigned, the transformation which relates the frames attached to neighboring links can be computed to solve the forward kinematics problem, i.e. finding the position and orientation of the manipulator hand relative to the base. Usually students' difficulties in visualization of 3D manipulator robots from written descriptions are related to the limited exposure to real manipulators. This expensive equipment is available in the university laboratories, but in a number not adequate for an effective training. Moreover, during lab assignments robotic movements are usually limited for safety issues and to reduce the risk of damaging robot body.

To overcome these difficulties, recent technologies, such as virtual reality and e-learning, can provide effective tools to support robotics courses. Virtual reality (VR) allows the students to visualize the robot in a three-dimensional environment in which they can freely navigate. They can interact with the robot, testing the effect on robot mechanics of modification in geometric parameters. Moreover, VR robots can be used to simulate programs before their actual execution on the real robot. E-learning can instead improve the accessibility to course material by the students, providing instruments that simplify the sharing and use of education tools among educators and students everywhere.

The training tool presented in this paper uses virtual reality to visualize robotic equipments in 3D. Its main goal is to increase the exposure of students to robot design and programming during their academic courses. A parallel aim of the 3D tool is to provide full portability and interoperability on several computer platforms and to be also available as an Internet application. In order to satisfy this goal, the system has been developed exploiting the Java3D APIs, a collection of Java classes providing a high-level object-oriented interface for the rendering of three-dimensional graphic programs. Java3D properties make it well suited for the simulation of virtual robots, as it provides facilities to represent geometry, event handling for the implementation of virtual sensors, and the possibility to dynamically change the scene graph to emulate robot motion [17].

The description of the environments and robots is provided to the tool through a set of XML (eXtensible Markup Languge) files [18]. An XML language has been developed to allow description of robot kinematics based on Denavit-

Hartenberg notation. The expressiveness of the language has been empirically assessed by modeling a number of heterogeneous robots including mobile robots, manipulators, and parallel kinematic chains.

In this paper, after a brief survey of 3D applications in robotics (Section 2), Section 3 introduces the format used to describe robotic scenarios for the Java3D-based modeling tool. The functionalities and implementation details are presented in Section 4. Finally, a description of tool applications is provided in Section 5.

2. 3D APPLICATIONS IN ROBOTICS

The application of 3D computer graphics technologies in robotics is continuously growing. Ranging from Internet robotics [8, 1, 12] to motion planning applications [10], the availability of a virtual environment to simulate robot movements before their actual execution is often regarded as highly valuable.

In spite of the large number of virtual reality tools, implementing new three-dimensional applications is still quite complex due to format dependencies, missing standards and lack of software engineering support, and requires expert knowledge to the programmer [3]. Most of the available tools for the development of 3D virtual scenarios for robotics lack generality because the design phase is strongly influenced by the problem peculiarities. Often, the models of the robots are buried in the code, preventing the modification of the virtual spaces without changes in the code itself. A general tool should instead allow a rapid and inexpensive reconfiguration of virtual robots. Moreover, it should allow a quick redesign of the environment to embody different sets of interconnections.

Several education tools based on virtual reality have been proposed in literature. Usually they are flexible robotic programming environments but they can often model only mobile robots [7, 9]. Tools that simulate manipulators show other drawbacks. The geometric representation of the link shape is often not supported [16] or limited to VRML [11]. Moreover, as the models of the robots are often buried in the code [1], the modification of the virtual spaces usually involves a change in the code itself requiring an insight knowledge of the program. Of course, commercial software packages propose complete solutions for virtual reality robotics [14] but this is usually an expensive possibility not suitable for university budgets.

The tool presented in the paper tries to overcome these shortcomings of available free robotics learning tools.

3. ROBOT REPRESENTATION

In order to guarantee an inexpensive reconfiguration of virtual robots and a quick redesign of the environments their descriptions are confined in external files. These files are used as input for the 3D tool, which must be able to parse the information to create objects in the virtual environment.

To hide the 3D modeling problem to the user, we decided to propose a new XML-based language specifically tailored to 3D robotic environments. The choice of a markup language is motivated by their common use in storage, transmission and exchange of information, as they allow the description in a standardized format of data or information contained in text. Moreover, XML has already proven to conveniently describe various types of structured data, as demonstrated by a growing number of XML-based languages in a wide range of domains. XML documents are human-readable, self-descriptive, easy to maintain while guaranteeing interoperability. In this specific context, the XML Schema technology [19] has been adopted to accurately define the structure, contents, and semantics of valid XML documents describing virtual robotic scenarios.

3.1 Representation of kinematic chains

Kinematic chains are described as sequences of links and information about shape and structure must be provided for each link. In particular, the kinematic properties of the links are expressed using the Denavit-Hartenberg parameters.

3.1.1 Denavit-Hartenberg parameters

Denavit and Hartenberg showed that the most compact representation of a transformation between two robot joints requires four parameters [4]. These parameters are currently the de-facto standard to describe a robot configuration (i.e., link lengths, joint positions, type of each joint). In the following we give a brief description of DH parameters, for a more in-depth discussion about DH and related concepts please refer to [2].

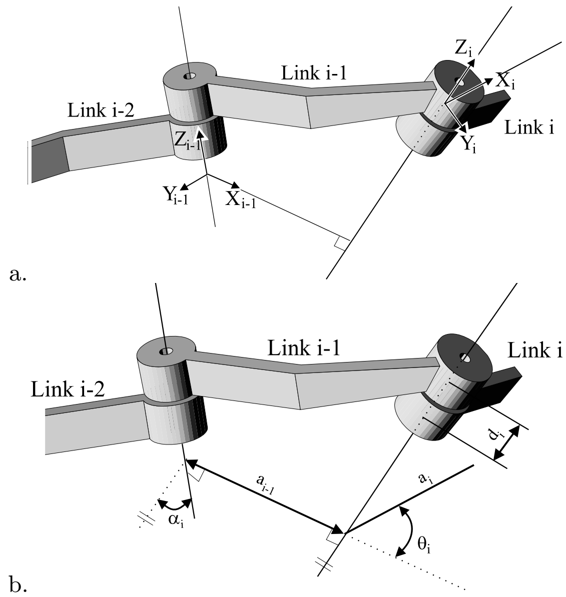


Figure 1: Joint description: a. Frame assignment, b. Denavit Hartenberg parameters

Figure 1 show two consecutive links of a robot manipulator. To describe the location of each link relative to the previous one, a frame is attached to each link (Figure 1-a). The position of $i-th$ frame is determined from $i-1$ frame by the use of the parameters a_{i-1}, α_{i-1}, d_i, and θ_i, where:

- a_{i-1} is the distance from Z_{i-1} to Z_i, the two joint axes, measured along X_{i-1}, the mutual perpendicular;

- α_{i-1} is the angle between Z_{i-1} and Z_i measured about X_{i-1};
- d_i is the distance from X_{i-1} to X_i measured along Z_i, and it is variable when the joint is translational;
- θ_i is the angle between X_{i-1} and X_i measured about Z_i. θ is variable when the joint is revolute, equal to 0 when it is translational.

Once the parameters have been assigned, the transformations that relate the frames attached to neighboring links can be concatenated to solve the forward kinematics problem.

3.1.2 An XML language for robot representation

The description of both the robots and virtual scenarios is provided to the modeling tool through a set of XML files, that can be automatically created by the Java3D tool given the robot parameters.

Robots can be either mobile robots, kinematic chains, sequence of kinematic chains or any system resulting from the composition of the previous ones. Mobile robots can be free-flying robots or mobile robots on a plane.

The geometry of the objects (static elements of the workspace, links of kinematic chains, or any other solid component of a robotic system) is described in a uniform way. A solid object is defined as a set of geometric shapes whose description is contained in separate files: only the file names appear in the XML document. As the geometric information are not directly included in the XML file, the language is guaranteed to be flexible and general: new file formats can be included when needed, i.e. the availability of new modeling tools supporting additional file formats does not require changes in the XML document. The tool currently accepts GTS (GNU Triangulation Surface Library), OFF (Geomview), and VRML formats, but Java3D loaders for other file formats are already available and easily integrable. A detailed and comprehensive specification of the developed XML language can be found in [6].

Figure 2: 3D model generated from the parsing of listing 1.

Table 1: Puma 560 Kinematic Parameters

Joint	$\alpha i - 1$	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	0	θ_2
3	0	a_2	d_3	θ_3
4	-90	a_3	d_4	θ_4
5	90	0	0	θ_5
6	-90	0	0	θ_6

As an example, Listing 1 shows an XML file describing a Puma 560, a popular six-d.o.f. manipulator (Figure 2) The file lists the sequence of links and for each link the following information are provided:

- two numbers defining its relative position in the chain,
- the Denavit-Hartenberg parameters with the range of the valid joint parameter values,
- the name of the files describing the shape of the link, and
- the frame transformation for the placement of the link, mapping the local frame of the geometric shape to the Denavit-Hartenberg frame.

Listing 1: XML file for the Puma 560 manipulator robot.

```
<?xml version="1.0" encoding="UTF-8"?>
<robot xmlns="http://www.mpml.org/robot"
xmlns:xsi=
       "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=
       "http://www.mpml.org/robot_robot.xsd">
<kinematicChainSequence>
 <kinematicChain>
  <link currentLink="0" precedingLink="-1">
   <DH alpha="0" a="0" minTheta="0"
       maxTheta="0" minD="0.63" maxD="0.63"/>
   <linkShape>
    <shape X="0" Y="0" Z="-0.67" rotX="0"
          rotY="0" rotZ="0">base1.off</shape>
    <shape X="0" Y="0" Z="-0.67" rotX="0"
          rotY="0" rotZ="0">base2.off</shape>
   </linkShape>
  </link>
  <link currentLink="1" precedingLink="0">
   <DH alpha="0" a="0" minTheta="-160"
       maxTheta="160" minD="0" maxD="0"/>
   <linkShape>
    <shape X="0" Y="0" Z="-0.67" rotX="0"
          rotY="0" rotZ="0">link1.off</shape>
   </linkShape>
  </link>

  ...

  <link currentLink="6" precedingLink="5">
   <DH alpha="-90" a="0" minTheta="-215"
       maxTheta="215" minD="0" maxD="0"/>
   <linkShape>
    <shape X="-0.41" Y="0.14" Z="0.24" rotX="180"
          rotY="0" rotZ="0">wrist1.off</shape>
    <shape X="-0.41" Y="0.14" Z="0.24" rotX="180"
          rotY="0" rotZ="0">wrist2.off</shape>
   </linkShape>
  </link>
 </kinematicChain>
</kinematicChainSequence>
</robot>
```

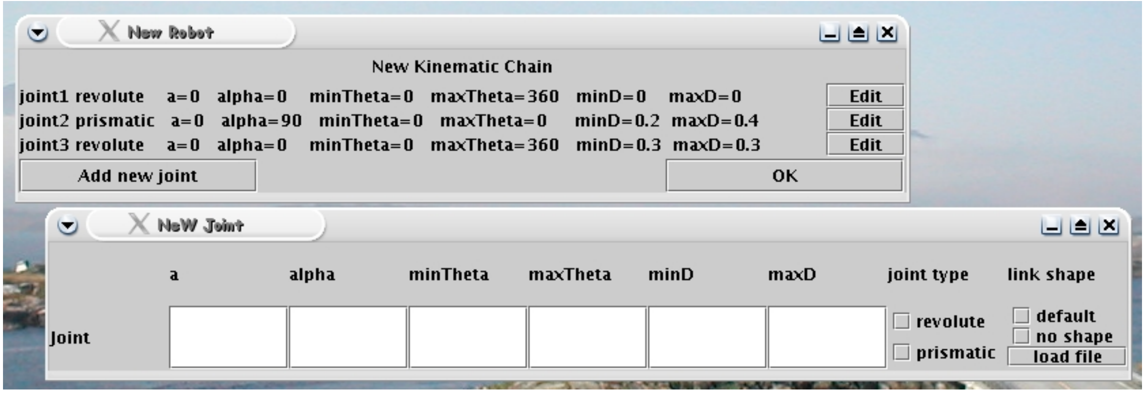


Figure 3: Java3D tool: input window.

4. THE JAVA3D-BASED TOOL

In this section, the functionalities and implementation details of the developed Java3D-based training tool are presented.

To design the robot, the user is provided with the input window shown in Figure 3. The Denavit-Hartenberg parameters of the robot, together with the link shapes, can be specified and an XML input file is automatically generated. The resulting robot is loaded in the tool and its graphical representation is shown in an appropriate window (Figure 4).

The graphical interface on the right introduces interaction commands through a set of menus, button, and sliders. Current implementation of the tool supports navigation functionalities such as change of viewpoint, zooming into/out of the scene, navigation into and rotation of the scene using the mouse. The user can change interactively the robot configuration modifying the values of its degrees of freedom either specifying a value in a text box or using the sliders. Finally, animations of robot motions can be displayed starting from a file containing a sequence of values for the degrees of freedom of the robots, usually a result of the execution of motion algorithms implemented for course assignments. This feature allows the graphically assessment of the correctness of the programs developed by the students before their actual execution on the real robots.

4.1 Implementation details

The tool has been implemented using Java3D, a standard extension to the Java 2 SDK, that provides a full integration with other Java components. Therefore, besides 3D graphics, all the modules of the application are implemented using the standard Java language.

Figure 5 details the overall structure of the system. The *XML Sax Parser* module receives the XML documents describing the virtual scenario to be rendered and parses them. The parser has been realized exploiting the SAX API [13]. SAX offers an event-driven parsing approach allowing a quick serial access to the document content in read-only mode. This approach has proven well suited for the needs of the 3D tool. The data are directly passed to the other modules that build the *scene-graph* corresponding to the environment to be rendered. Moreover, the parser automatically tests the well-formedness of the XML documents and validates them against the associated XML schema, relieving the 3D tool from the burden of onerous checks to guarantee data consistency and coherence.

Parsed information is passed to the *Geometry Loader* and *Robot* modules. The *Geometry Loader* module reads the geometric information contained in the files describing the shape of the elements in the scenario and sets up the Java3D objects required by the rendering engine to display geometry. This module is based on the Java3D *Loader* interface that provides a straightforward way to deal with the different existing 3D file formats. Current implementation of the tool supports triangulated (gts, off, and vrml) and constructive solid geometry formats, but it can be easily extended as a growing number of loaders are made freely available on the Web.

The *Robot* module builds the *scene sub-graph* that models the robotic system. Thanks to the use of XML, virtual scenario description is fully decoupled from the tool internal representation. A notable difference with other tools [1] is that the user is not required to deal directly with, nor to be aware of, the 3D representation of the scene. Moreover the tool has proven general enough to support a wide variety of different robotic systems.

The *3D Object Display and Animation* module, exploiting information available in the *Robot* module, sets up the *scene-graph* to model the complete 3D space and a set of virtual instruments to interact with it. The user commands interacting with the 3D scene are passed to the *3D Object Display and Animation* module by the *GUI* module that sets up a panel with menus, buttons and sliders to supports the functionalities previously described.

5. APPLICATIONS

The tool capabilities to represent the robot from its Denavit-Hartenberg (DH) parameters make it suitable as an educational tool for robotics courses. Solving the problem of frame assignment according to the DH representation is usually a challenging exercise for students. The tool, giving a 3D representation of the robot, can be used by the students to autonomously validate their solutions resulting in a speed

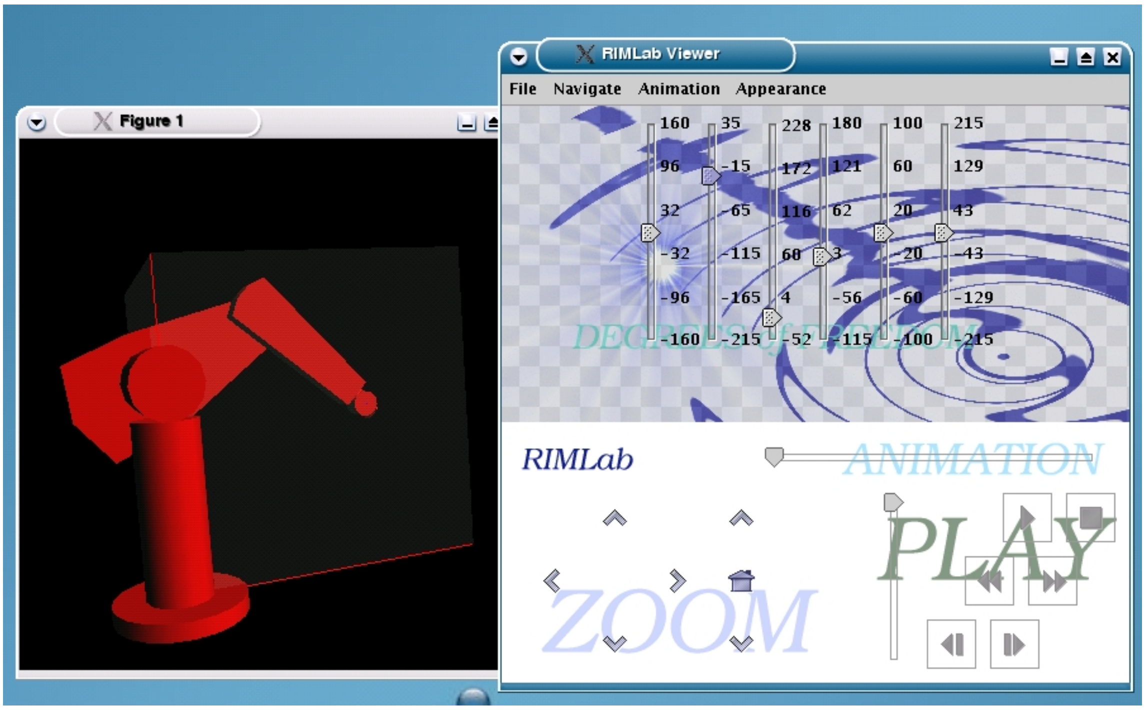


Figure 4: A screenshot of the Java3D tool.

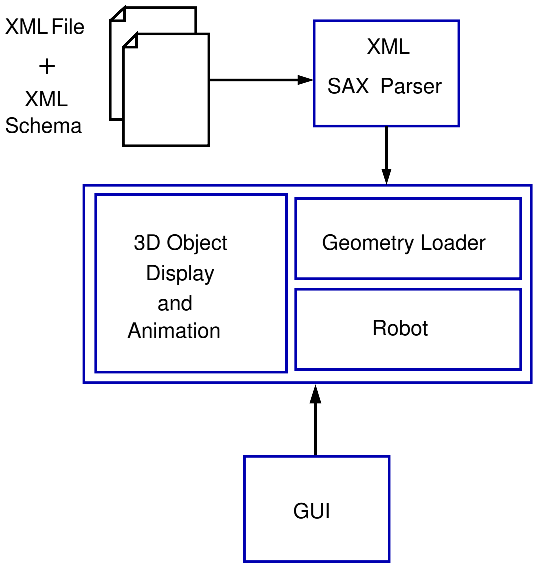


Figure 5: Overall structure of the tool.

up in the learning process. Moreover, the availability of the sliders to interactively change the degree of freedom values of the robot provides an instrument for the study of forward kinematics.

The 3D tool is also valuable in the analysis of robot workspace when alternative kinematic arrangements are investigated. The student can interactively change the robot configurations and test them for collisions, using kinematic and geometrical models of a robotic system. Figure 6 shows the reachable workspace of a system consisting of a central vertical mount of a Manus manipulator on top of a Nomad 200 mobile robot.

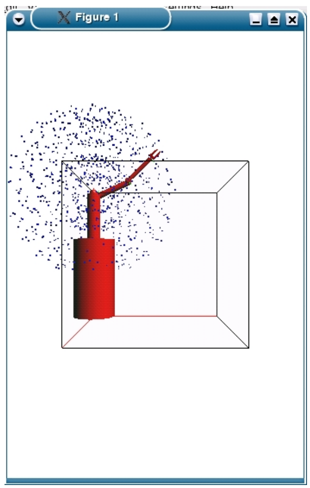


Figure 6: Mobile base-arm integration and reachable workspace with a vertical mount of the arm on top of the base.

5.1 Research applications

The tool can be used as a starting point for the development of more advanced systems. So far, it has been successfully exploited for the implementation of a graphical interface for motion planning applications and for trajectory representations.

In the former, the path computed by the planner, which contains the sequence of degrees of freedom values, is rendered by the tool as an animation of the robot motion. Regarding trajectory representations, the tool has been interfaced with a six degree of freedom Polhemus tracker with the goal of qualitatively evaluating control algorithms for mobile and manipulator robots.

6. CONCLUSIONS

Virtual reality can be a valuable resource for the implementation of applications for teaching and training purposes in robotic courses. In this paper we have described an open-source, freely available tool for the creation of virtual environments and robots to support students in the learning process. The early version of the tool has been already used in a robotics course demonstrating its usefulness. A quantitative analysis of its suitability in reducing student learning difficulties is planned in the next months through the use of subjective tests.

The system, built upon open standard and Java technologies, has proven fully portable across several platforms. The tool is freely available for download at `http://mpb.ce.unipr.it` upon request to the authors.

7. REFERENCES

[1] I. Belousov, R. Chellali, and G. J. Clapworthy. Virtual reality tools for internet robotics. In *IEEE Int. Conf. on Robotics and Automation*, pages 1878–1883, Seoul, Korea, 2001.

[2] J. Craig. *Introduction to robotics*. Addison-Wesley, 1989.

[3] R. Dachselt. Contigra - towards a document-based approach to 3D components. In *Workshop "Structured Design of Virtual Environments and 3DComponents "*, 2001.

[4] J. Denavit and R. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 22, 1955.

[5] European Robotics Forum and European Robotics Research Network. A white paper on the status and opportunities of the european robotics industry. Available at: `http://www.euron.org/`.

[6] E. Fantini and M. Reggiani. A Practical Introduction to the Motion Planning Markup Language (MPML). Technical report, Robotics and Intelligent Machines Laboratory (RIMLAB), Università degli Studi di Parma, 2003.

[7] E. Guimaraes, A. Maffeis, J. Pereira, B. Russo, E. Cardozo, M. Bergerman, and M. Magalhaes. REAL: a virtual laboratory for mobile robot experiments. *IEEE Trans. on Education*, 46(1):37–42, 2003.

[8] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Sensor-based space robotics -rotex and its telerobotic features. *IEEE Trans. on Robotics and Automation*, 9(5):649 – 663, 1993.

[9] R. Kuc, E. Jackson, and A. Kuc. Teaching introductory autonomous robotics with JavaScript simulations and actual robots. *IEEE Trans. on Education*, 47(1):74–81, 2004.

[10] J. Kuffner, S. Kagami, M. Inaba, and H. Inoue. Graphical simulation and high-level control of humanoid robots. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 1943–1948, Takamatsu, Japan, 2000.

[11] A. Lobov, J. Lastra, and R. Tuokko. A collaborative framework for learning robot mechaniscs: RIO-robotics illustrative software. In *ASEE/IEEE Frontiers in Education Conference*, pages F4E–12–F4E–16, Boulder, CO, 2003.

[12] R. Marin, P. Sanz, and A. del Pobil. A predictive interface based on virtual and augmented reality for task specification in a web telerobotic system. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 3005–3010, Lausanne, Switzerland, 2002.

[13] D. Megginson. The Simple API for XML (SAX). `http://www.saxproject.org`.

[14] O. Michel. Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.

[15] MIT Course. 6.270 autonomous robot design competition. `http://web.mit.edu/6.270/`.

[16] M. Robinette and R. Manseur. ROBOT-DRAW, an Internet-based visualization tool for robotics education. *IEEE Trans. on Education*, 44(1):29–34, 2001.

[17] N. Smith, C. Egert, E. Cuddihy, and D. Walters. Implementing virtual robots in java3d using a subsumption architecture. In *AACE World Conference on the WWW and the Internet*, pages 975–980, Honolulu, HI, USA, 1999.

[18] World Wide Web Consortium. XML. `http://www.w3.org/XML/`.

[19] World Wide Web Consortium. Xml schema. `http://www.w3.org/XML/Schema.html`.

External Interaction Management of VRML Scenes for E-Learning Applications

José M. Cubero✉, Luis Salgado✉, Jesús Bescós✉, Francisco Morán✉ and Julián Cabrera✉

Grupo de Tratamiento de Imágenes

✉ETS. Ing. Telecomunicación
Universidad Politécnica de Madrid
28040 Madrid - Spain
+34.913.367.353

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

✉Escuela Politécnica Superior
Universidad Autónoma de Madrid
28049 Madrid - Spain
+34. 914.972.285

Jesus.Bescos@ii.uam.es

ABSTRACT

This paper describes an innovative approach to solve some of the problems that arise when integrating virtual reality capabilities into e-learning environments. The VRML representation of a scene includes, along with its geometric description, a full specification of the student-scene interaction logic. This representation is rendered by a browser, which also orchestrates the interaction according to the logic. Such a mechanism implies reprogramming and/or replicating partly the logic when modifying the interaction scheme of a single scene for different students. It also prevents any external access to student's actions or scene reactions, which is necessary for on-line evaluation or instruction. We propose to expand the standard interaction mechanism of VRML so that both the specification of the scene logic and the interaction flow are managed by an external and centralized entity following a client-server approach, hence solving the identified problems, while additionally increasing design efficiency and content protection.

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*; K.6.4 [**Management of Computing and Information Systems**]: System Management - *Centralization/decentralization*; C.0 [**Computer Systems Organization**]: General - *System architectures*.

General Terms

Algorithms, Management, Performance, Design, Human Factors.

Keywords

Virtual environments, e-learning applications, external interaction, information management, client-server architecture, adaptability, VRML, Java.

1. INTRODUCTION

The current development of Internet applications, and especially of those based on computer graphics, allows for the use of e-learning systems including not only multimedia information but also Virtual Reality (VR) capabilities. Thanks to VR, a student can believe to be transferred to another place and enjoy interactive and immersive experiences, which can greatly improve her/his learning process. Low-cost VR experiences are possible with publicly available plug-ins for common web browsers, which can interpret and render interactive 3D scenes described in standard languages such as VRML [8][6] and X3D [9].

When immersed in a virtual world, a student experiences a scene which follows a behavior pattern established by the objectives imposed in the learning process. This pattern is implemented via an interaction logic controlling the interaction flow between the student and the virtual world, and therefore must be programmed and adapted for every new learning situation.

3D scene modeling languages, like VRML, allow to code that interaction logic in the same file where the geometric description of the scene is defined. The interaction management is usually event-based, an event being just some kind of basic information message being 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 just routed to other active nodes as a basic means to modify their characteristics (e.g., their position). In order 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 the corresponding output events, which are then routed to the desired nodes. These routing and scripting mechanisms, which conform the interaction logic implementing the scene behavior pattern, result in a traffic of events [12] that contains all the information related to user-scene interaction. This information flow is established between the student and the browser that interprets and renders the coded scene.

This scheme results in a lack of flexibility, from at least two points of view. First, every desired modification of the behavior 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 can-

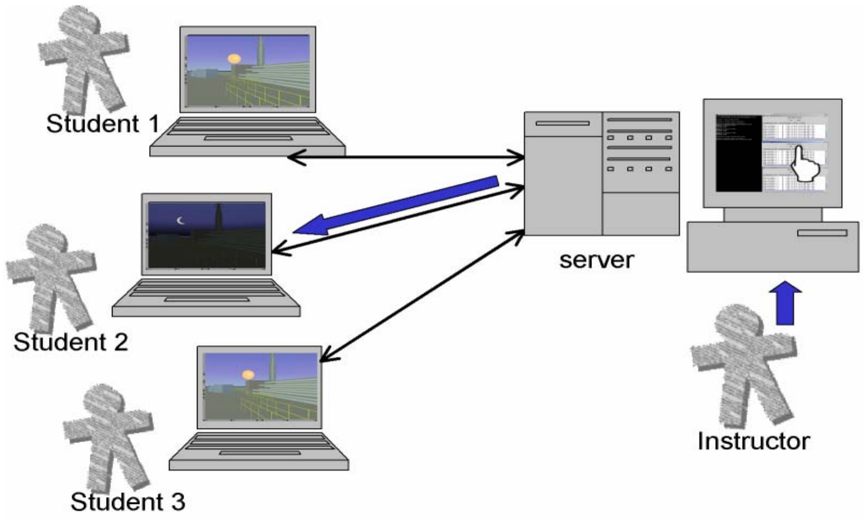


Figure 1. Example of modification of the student virtual environment conditions

not 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 in response to the student's actions, which is one of the main features that e-learning technology 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 [1]. We propose to deepen into this approach and solve all the identified problems via the inclusion of a client-server scheme not just intended to centralize course management (an architecture frequently used in e-learning systems [12][2][3]) but specifically targeted at the management of the interaction information flow. This innovative approach completely decouples the behavior pattern of the scene from its geometric description, hence reinforcing the reusability of the models, which may be linked to different logics, as well as the flexibility in the design, as parallel developments of the geometry and the logic can be carried out almost independently.

Moreover, 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 behavior pattern, which is kept at all times in the server, so the overall scene is inherently protected against undesired copies or misuse.

Additionally, our approach enables to achieve on-line inspection and adaptation of the interaction flow between several students (i.e., several clients), each having her/his own virtual learning environment independent from the others, and a single evaluation and instruction center (i.e., the server). An example is shown in Figure 1. Three students are working on the same exercise developed in a specific virtual environment. In a specific moment, the student number 2 shows a higher level of experience. Then, the instructor, located at the server machine, decides to increase the difficulty of his/her exercise changing on-line the illumination conditions of the scene, from daylight to nightlight. The student number 2 continues his/her exercise in those conditions while the rest of the students develop the exercise with the initial illumination. Our approach allows this kind of on-line interaction.

The architecture 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) [11]. Section 2 summarizes the VRML standard interaction mechanism and Section 3 describes the overall architecture proposed to modify it. Section 4 delves into some interesting implementation details and Section 5 describes the performed tests. Finally, Section 6 concludes the paper.

2. VRML INTERACTION MANAGEMENT

The VRML standard mechanism for managing the interaction between the user (i.e., the student, in an e-learning application) and the virtual scene can be summarized in three steps:

- Identification and unique labeling of the scene objects that will provide interaction capabilities.
- Definition, for these objects, of the input events they are sensible to and/or the output events they generate.
- Establishment of routes between pairs of objects via linking the input events of one with the output events of another. According to these routes of interaction, the VRML browser delivers properly each generated output event to the object designed to receive it. The receiver object can be directly modified using the information provided by the output event but, to simulate complex situations, a script can process the original output event and yield one or more output events based on it. The system described here exploits this mechanism by the use of the Scripting Authoring Interface (SAI).

The management of a VRML scene generates events continuously. VRML does not distinguish between discrete events, such as those generated as the user activates a *TouchSensor* node, and continuous events, i.e., those generated by sampling over time conceptually continuous variables, such as those generated by a position sensor, a time sensor or the user's action on mobile objects, typically registered by the so-called "drag sensors" (*CylinderSensor*, *PlaneSensor* and *SphereSensor*). For these continuous events, an ideal implementation of a VRML browser would generate infinite samples. Actually, the sampling frequency depends on the particular implementation of the browser, and typically matches the number of rendered frames per second. In most applications, the number of such events is much more than needed to control the user's actions [12], so there is a need to filter them out to improve significantly the efficiency of the final system.

to exchange events from the VRML browser, where the scene is rendered, to the VRIM and vice versa.

The EIM procedures are called from a VRML *Script* node of the scene. When the scene is rendered, the EIM establishes a TCP communication with the EIS, which should be available via an IP address (or a valid DNS name) defined by an input parameter, also referenced in the VRML *Script* node. Every event routed from the scene to the EIM is processed by this module and sent to the EIS through TCP messages that follow the EIP. When the browser is closed or another VRML scene is loaded, the EIM detects it and automatically shuts down the TCP connection with the EIS.

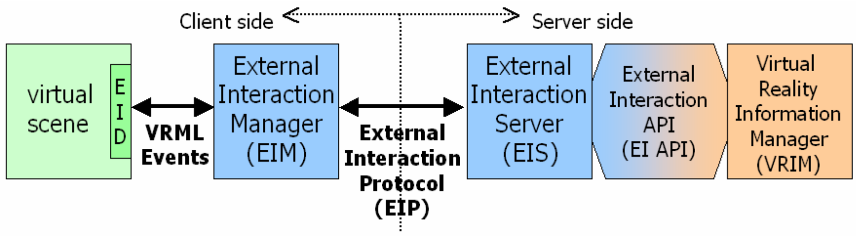


Figure 2. Interaction Management System overview

3. SYSTEM ARCHITECTURE OVERVIEW

Figure 2 presents a block diagram of the proposed client-server Interaction Management System (IMS) architecture. It allows user-scene interactions to be managed by an external entity called Virtual Reality Information Manager (VRIM), which implements the virtual scene interaction logic. It typically runs on a remote server and should operate in real time, network bandwidth permitting.

The IMS implements a bidirectional event-based communication 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 is a module that, on one side, handles the exchange of VRML events with the scene through the External Interaction Definition (EID) SAI and, on the other, establishes a TCP connection with the EIS, which later follows the External Interaction Protocol (EIP). The VRML events are hence 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 overview of these interfaces, modules and protocols. Section 4 gives a more detailed description of the modules, of which most have been implemented in Java to ensure portability and to take advantage of the Java support offered by VRML.

External Interaction Definition

The interaction, that is managed outside of the virtual scene, has to be defined anyway according to the VRML standard mechanisms introduced in Section 2: it is necessary to specify which objects will be subject to the external interaction, and which type of information they will either 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 External Interaction Definition (EID) in Figure 2. It consists of the list of the VRML events of the objects allowing external interaction management, and of several parameters required for the proper operation of the EIM.

External Interaction Manager

The External Interaction Manager (EIM in Figure 2) is the script in charge of establishing a TCP connection with the EIS in order

External Interaction Server

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

External Interaction Protocol

The External Interaction Protocol (EIP) regulates the communication of asynchronous messages between the EIM and EIS to allow the transmission of events produced by the user interaction within the virtual scene to/from the VRIM.

External Interaction API

The External Interaction API (EI API) is the interface between the EIS and the VRIM that allows an independent and possibly parallel implementation of both modules. The EIS functionality (i.e., manage the TCP communication) is completely independent from the particular virtual scene, whereas the VRIM implementation fully depends on it, as it 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.

Virtual Reality Information Manager

The Virtual Reality Information Manager (VRIM) implements the behavior pattern of the scene. It is able to manage simultaneously the information provided by different instances of a scene being rendered in different clients and interacted upon by different users.

The VRIM usually receives VRML events related to the user-scene interactions through the EIM, EIS and EI API, and sends back VRML events in response and through the reverse path. But the VRIM is also able to generate itself messages (events) to be sent to the user at any time, e.g., upon the orders of an instructor.

4. DETAILED MODULES DESCRIPTION

4.1 External Interaction Events

To handle the different kinds of information, the generated External Interaction Events (EIE) are classified into three groups:

- Location Events, which indicate the position of the avatar, representing the user in the scene rendered in the client.
- Timing Events, aimed at transmitting temporal information of the client where the scene is being rendered.
- External Operation Events, which group the different VRML events generated by the user interaction with specific objects in the scene, together with the input events that will be sent by the VRIM as a response to the user's actions, modifying the appearance of the objects being rendered.

The first two types are intended to be used for tracking purposes, 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.

Communication with the EIS

The ClientTCP module establishes a TCP connection with the EIS. The IP address (or a valid DNS name) of the EIS is an input parameter obtained from the EID.

Filtering of Continuous Events

For most applications, the number of continuously generated events described in Section 2 is excessive to accurately control the user's interaction with the scene. The EIM design allows to configure the number of such events in order to the application requirements, thus reducing significantly the system communication overhead. The EIM sends events at intervals not less than a minimum value specified in the scene through the EID. For instance, the EOperationMgr module (see Figure 3) filters the *SFVec2f, SFVec3f* and *SFRotation* VRML events continuously generated by the *CylinderSensor*, *PlaneSensor* and *SphereSensor* VRML nodes, respectively.

The continuous Location Events generated by the user's motion are filtered by the LocationMgr module and, as the EIM knows if the scene is being rendered, it generates the Timing Events and filters them with the TimingMgr module.

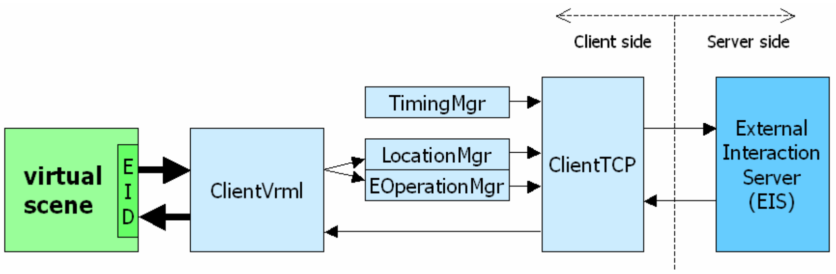


Figure 3. Block diagram of the External Interaction Manager

The last one helps tracking the user's interaction with specific objects in the scene, and to route the corresponding responses provided by the VRIM which modify the scene appearance according to the specific implementation of the behavior pattern.

4.2 External Interaction Manager

A detailed block diagram of the EIM is presented in Figure 3. The path followed by the different event types is identified through the arrows communicating the different modules, the thick ones representing VRML events and the thin ones the three kinds of EIE (Location, Timing and External Operation Events).

Interface with the VRML Scene

The ClientVrml module acts as the interface with the VRML browser. On one side, it extracts the input parameters and the VRML events generated by the user's interaction in the virtual scene. On the other, it updates the scene with the information provided by the External Operation Events received from the VRIM, typically as a response to some previous incoming event originated at the client, but possibly also as an event generated by an instructor behind the VRIM.

4.3 External Interaction Server

Figure 4 shows in detail the different modules that form the EIS. The path followed by the different kinds of events is identified through the arrows communicating the different modules.

The operation of the EIS can be described as follows.

The TCP server (EIServer in Figure 4) is set up initially to accept connections from the client (thick and continuous arrow).

When a new client connection is requested, the server launches a Client Manager (ClientMgr or simply CM) which handles the communication with a particular user. Every CM keeps waiting for the incoming events generated by the client (EIE from client) to which it is connected.

Whenever an event arrives, a specific Event Manager (EventMgr or simply EM) is created to process it by calling the appropriate set of methods of the EI API: Location or Timing or External Operation API.

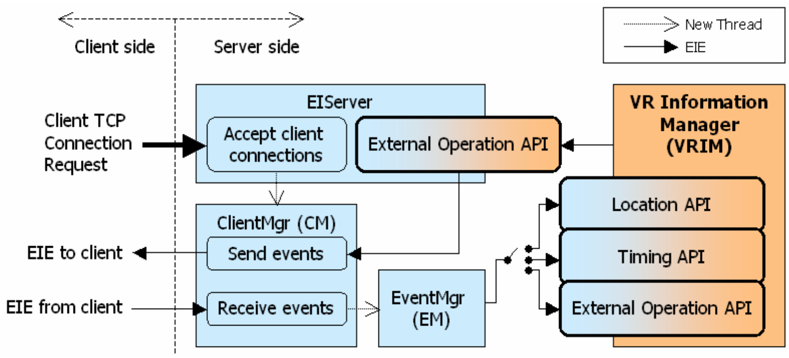


Figure 4. Diagram of the Java classes of the External Interaction Server

On the other side, the events received from the VRIM are passed to the appropriate CM, which forwards them to the client (EIE to client).

4.4 External Interaction API

The Location and Timing Events can only be used in a unidirectional way: from the scene to the VRIM, as they provide information solely depending on the user's operations at the client side. Therefore, the Location and Timing API must be unidirectional.

On the other hand, the External Operation Events do usually require bidirectional communication between the scene and the VRIM, as the VRIM uses this type of events to control the operations performed by the user in the scene (incoming events), and to send back modifications to the scene (outgoing events). Consequently, the External Operation API must be bidirectional, and appears twice in Figure 4, where the two paths of the incoming/outgoing events have been separated for the sake of clarity.

4.5 Virtual Reality Information Manager

The VRIM implements a method for every function of the EI API accessible from the EIS, i.e., the Location API, the Timing API and the External Operation API. These are the methods that finally implement the behavior pattern of the virtual scene, i.e., the number and type of events that will be generated as a response to the user's action or situation. This module, located at the server, compiles the interaction logic that, with VRML standard mechanisms, would have to be integrated either into the scene description file or in attached scripts, that would have to be located at the client before any rendering of the scene could occur.

Although the VRIM can be programmed using Java as the rest of the modules described above, it would perhaps be preferable to use other more efficient high level programming languages.

5. EFFICIENCY EVALUATION

As in any client-server architecture, the efficiency issues related to the number of clients that can be supported by a single server are particularly relevant. In the case of our system, efficiency tests have been carried out in which the computational load introduced by the IMS has been evaluated by measuring the ratio of the processor clock cycles used by the IMS at the server to handle the correct transmission of events.

Two main system variables were found to affect significantly the IMS server efficiency and have been considered during the tests: the number of clients to be supported, and the continuous event filtering parameter, i.e., the minimum interval between continuous events transmitted from each client to the server. In normal operation, user interactivity hardly generates a relevant number of discrete events compared with the number of continuous events (in particular, Location or Timing Events, which can be generated and transmitted as much as one every rendered frame).

Two different scenarios have been considered. In the first one, a predefined tour has been implemented in which the avatar is being moved continuously within a simple virtual scene. In this case, Location Events are continuously transmitted to the server. In the second, an automatic generator of External Operation Events has been implemented to simulate a user's interaction with an object generating continuous drag sensor events.

The overhead introduced by the proposed architecture was measured at both the client and server ends. As expected, the one imposed by the EIM at the client side was meaningless, while the one of the EIS at the server side could be really significant depending on the values of the system variables described above.

Table 1. Ratio (%) **of processor clock cycles used by the server**

		Number of clients							
		1	2	3	4	5	6	7	8
LEI (ms)	0	1	8	10	11	13	17	21	30
	40	1	2	3	4	5	7	8	10
	500	0	0	1	1	1	1	2	2
EOEI (ms)	0	8	15	30	50	BLOCKED			
	40	2	5	8	10	13	15	18	22
	100	1	3	5	6	8	9	10	12

The percentage of the processor clock cycles used by the server with the two scenarios was measured on a PIV@2.66 GHz PC with 512 MB of RAM, and is shown in Table 1. The first three rows correspond to the first scenario, where the LEI value is the Location Event (minimum) Interval in milliseconds. The last three rows correspond to the second scenario, and EOEI stands for External Operation Event (minimum) Interval.

As it can be observed, the main limiting factor is the EOEI, the overhead imposed by the Location Events being significantly lower for any number of clients. Therefore, as expected, it is mandatory to filter the External Operation Events. However, it has to be considered that these tests assume that the users at the different clients are generating non-stop continuous events, something that, apart from Location and Timing Events, never happens in real applications. According to the 12% in the rightmost, bottom cell of Table 1, it is clear that many more than ten clients can be supported by a single server with the correct transmission of ten External Operation Events per second (i.e., one every 100 ms) under this artificially extreme situation. Therefore, it can be expected that the required frequency of event transmission be met in real applications.

6. CONCLUSIONS

The Interaction Management System (IMS) we propose allows for the external management of the interaction-related information generated by the actions of a student in a virtual scene, separating the logic ruling the scene behavior from its geometric description. An entity completely external to the scene, the VR Information Manager (VRIM), is able to monitor at a remote server the actions the student performs at the client, and manages remotely the behavior of the scene, modifying it when necessary.

The IMS helps to flexibly and efficiently model and program interactive 3D scenes. Modifications of the logic of the scene, which resides in the VRIM, do not imply modifications in its geometric description. This enables transparency for the student and parallel developments of the geometry and the logic of the virtual environment. The IMS is modular, with well defined interfaces that allow easy modifications and evolutions of the system.

Tests developed on the IMS show its correct functioning even in stressful situations, i.e., with too short a minimum interval between External Operation Events, which is the determinant factor for the computational load at the server. In real applications, with a reasonable number of clients and a normal value for that minimum interval, the correct transmission of the number of events required to effectively control the behavior of the scene can be guaranteed.

The architecture we propose is being used in the EU-funded research project "Self Learning Integrated Methodology –Virtual Reality Tool" (SLIM-VRT, IST-2001-33184) [11], where an e-learning application in a shipping environment is under development. The particularities of this application have been addressed on a case study based approach where the flexibility, reusability and efficiency provided by the separation of the geometric description of the 3D scenes and the management of their logic are proving to be extremely useful.

Future developments are foreseen in the following directions:

- Improved event filtering: currently, only the VRML events generated by drag sensors are filtered, but it could probably be advantageous to let the scene logic designer to add other events to be filtered, and perhaps even to establish priorities among events.
- Porting to X3D: the information types have been defined to be as general as possible in the IMS, so an evolution to support scenes described in X3D should only imply modifying the interface with the scene (i.e., the ClientVrml module in the EIM), the rest of the system being, in principle, independent of the language used to describe the scene.

7. ACKNOLEDGEMENTS

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] Bouras, Ch., Philopoulos, A., and Tsiatsos, Th. *A Networked Intelligent Distributed Virtual Training Environment: A first Approach*. International Workshop on Intelligent Multimedia Computing and Networking (IMMCN 2000, a constituent of JCIS 2000), 2000.

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

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

[4] Eckel, B. *Thinking in JAVA*. Prentice-Hall, 2000.

[5] Harold, E. R. *JAVA Network Programming*. O'Reilly, 2000.

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

[7] Hughes, M., Shoffner, M., and Hamner, D. *JAVA Network Programming*. Manning, 1999.

[8] ISO/IEC 14772-1:1997. *Information technology - Computer graphics and image processing - The Virtual Reality Modeling Language. Part 1: Functional specification and UTF-8 encoding*. ISO/IEC, 1997.

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

[10] Qingping, L. et al. *A Novel Method for Supporting Collaborative Interaction Management in Web-based CVE*. Proceedings of the ACM symposium on virtual reality software and technology, 2003.

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

[12] 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, Vol. 13, No. 3, June 2004.

Web Based Delivery of 3D Developmental Anatomy

Harry Brenton[1 2], Juan Hernandez[1], Fernando Bello[1], Paul Strutton[1], Tony Firth[1], Ara Darzi[1]

[1]Imperial College London
[2]St Mary's Hospital, Dept. of Surgical Oncology and Technology
Praed Street, London W2 1NY
+44 (0)794 114 5615
h.brenton@imperial.ac.uk

ABSTRACT
Web-based 3D has proven educationally beneficial for simulating surgical procedures over the web. This paper describes an ongoing project to apply a similar approach to the modeling and delivery of developmental anatomy and pathology. Standard imaging techniques failed to provide good images for 3D reconstruction, for this reason a new modeling technique was developed using a commercial software package (Alias/Wavefront *Maya*).

Categories and Subject Descriptors
K.3.1 [**Computing Milieux**]: Computer Uses in Education – *computer- assisted instruction, distance learning*.

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *animations, artificial, augmented and virtual realities*.

I.3.7 [**Computer Graphics**]: Three Dimensional Graphics and Realism – *virtual reality, animation, color, shading, shadowing and texture*.

General Terms
Design, Experimentation.

Keywords
Anatomy, web, 3D, medical education, development, Maya, modeling, teaching, learning.

1. INTRODUCTION
There is currently considerable debate about the educational benefits of using 3D models as an aid for teaching anatomy to medical students. Peninsula Medical School has led the way in the UK by replacing the actual dissection of cadavers with an approach that relies heavily upon computer imaging [9]. Others argue that computer models should supplement rather than supplant dissection [10]. However, both sides agree that 3D imaging is an increasingly useful tool within medical education. Of particular educational benefit is the ability to show the spatial relationship between structures as they develop over time. This paper presents a new approach to 3D modeling for use in developmental anatomy and pathology, exploring its benefits and discussing potential future developments.

1.1 Educational benefits
The key benefits of 3D modeling for anatomy and pathology teaching are:

- The clear demonstration of spatial relationships between anatomical structures.
- The provision of a narrative timeline which enables the viewer to see key stages of anatomical development, aided by controls for pause, rewind and fast forward.
- The ability to see the progression of a disease over time: often the spread of cancer is not evenly spaced.
- The provision of flexible learning pathways which allow students to access the amount and type of information to suit their personal needs.
- The ability to disseminate a learning resource over an intranet or CD ROM using standard Flash and Java plug-ins.

In contrast, two-dimensional diagrams and photographs do not show spatial relationships and they present a static snapshot in time which poorly communicates changes in development. Magnetic Resonance Imaging (MRI), Computerised Tomography (CT) and X-rays are anatomically accurate, but require expertise to interpret, may lack the needed level of detail and present a static moment in time.

1.2 Case study: the brachial plexus
The embryological development of the brachial plexus was chosen as a suitable example to explore new approaches to 3D modeling and delivery. The brachial plexus is a network of nerves originating from the spinal cord, emerging between the vertebrae in the neck and extending down to the shoulder, arm, hand, and fingers. Brachial refers to the arm, plexus means network.

Study of the nervous system forms an important component of the undergraduate medical curriculum. Students are expected to:

memorise the structure of the brachial plexus; understand how the brachial plexus relates to the peripheral nervous system and the anatomy of the upper limb; develop an ability to clinically evaluate how nerve injuries affect the upper limb.

Undergraduate medical students find the structure of the brachial plexus very hard to remember and have referred to it as "like trying to memorise Clapham Junction" (the busiest railway station in the UK). Their usual approach is superficial rote learning from two-dimensional diagrams which encourages a surface understanding of the subject. Two-dimensional representations are of limited educational value because the nerve roots and fibres grow to form a complex arrangement within a three-dimensional space.

3D modeling aims to help students gain a deeper understanding the formation of the brachial plexus and how the nerves come out of it to innervate (supply sensation) to the whole limb. A good understanding of these concepts provides an educational scaffold from which to develop the diagnostic skills necessary to infer the consequences of trauma or disease.

The embryological development of the brachial plexus is the subject of a lecture given by a professor of anatomy at Imperial College London. The lecture represents an original approach to the subject and it forms the 'storyboard' for the creation of the electronic learning resource.

2. MODELING

2.1 Standard techniques

The standard technique for building 3D models of human anatomy involves a number of stages: scanning, enhancement, segmentation, and volume or surface rendering.

2.1.1 Scanning

A coordinated series of 2D images is acquired using a scanning technique such as CT or MRI. Each image shows a slice through the body at regular intervals, typically one 1mm apart. The type of imaging modality used can be varied to suit the requirements of the disease/clinical indication. For example, CT is often used to image colorectal cancer because it presents a detailed picture of how far the cancer has spread to areas surrounding the colon. MRIs are often used to plan neurosurgery because they are good at imaging vital structures in soft tissues which need to be avoided when removing nearby tumours.

2.1.2 Enhancement

The raw images are digitally manipulated to enhance image quality. The process is analogous to adjusting the tonal range of a photograph in *Adobe Photoshop* using the histogram tool. Common techniques include contrast enhancement, noise reduction and interpolation. Enhancement must be done with great care as it can emphasise image artifacts and even lead to a loss of information if not correctly used.

2.1.3 Segmentation

Vector drawing tools are used to trace chosen regions. For example, segmenting the bones in an MRI of a knee involves drawing an outline around the femur and tibia to visually isolate them from surrounding cartilage and muscle. This process also has the effect of electronically tagging the bones so they can be uniquely addressed and manipulated.

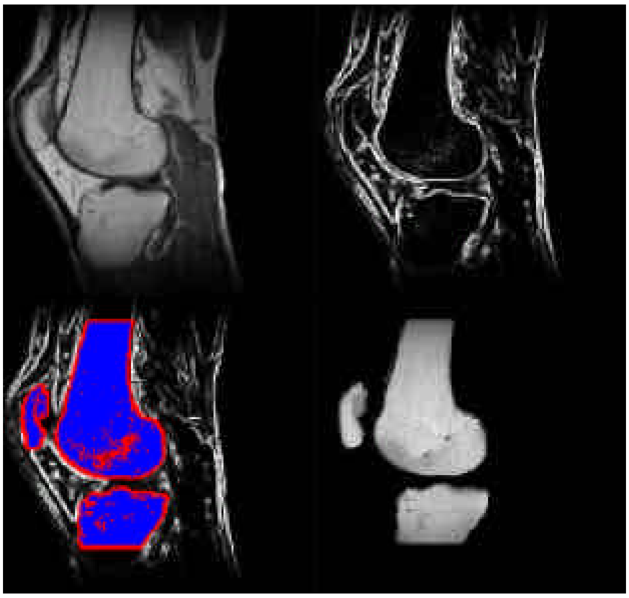


Figure 1. Segmentation of an MR knee image

Without segmentation unwanted regions may get included when the series of 2D images is reconstructed in 3D (*figure 2*). Segmentation can be done manually but it is a slow and laborious task. Automated or semi-automated segmentation software [2] can help speed up the process by using various techniques such as pattern recognition to help extract the border of structures.

2.1.4 Volume or surface rendering

Once segmentation is complete, the software processes each 2D slice in turn to build a 3D volume dataset. Volume data is suitable for medical images because it shows the inside of solid objects and allows outer layers to be cut away. However, real-time interaction is limited because each new view must be fully recomputed, requiring substantial processing power.

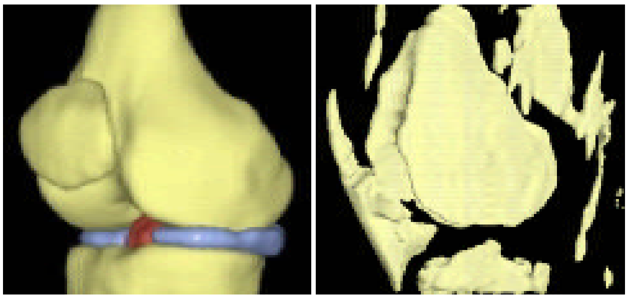


Figure 2. 3D reconstruction i) with previous segmentation ii) without segmentation

A common way of extracting polygonal surface information is to use the 'marching cubes' algorithm [7] which tests each voxel in turn in order to locate the boundary of a wanted object within a 3D array of voxel values. The data are then used to construct a numerical description of the surface. Because the surface is computed only once the viewpoint can be rapidly updated allowing for fast interaction. However, it is not possible to cut away outer layers to show the inside of solid objects. Once a surface dataset is created it can be conventionally texture mapped using software such as the *Visualisation Toolkit* (*VTK*)[13].

2.2 Level of detail

Careful judgment is required to assess the appropriate level of detail required for different applications. A neurosurgeon preparing to operate on a brain tumor requires a high-fidelity model showing the soft tissue of a specific patient as accurately as possible. In contrast, a medical student learning basic brain anatomy requires a only simplified generic model to illustrate key learning points. Artistic as well as technical skills are required to balance anatomical accuracy with the need to clearly convey medical concepts.

2.3 New techniques for segmentation and modeling

2.3.1 Segmentation with Macromedia Fireworks

The vector tools in *Macromedia Fireworks* were used to trace the outline of a growing limb from a series of human embryo MRIs showing 10 key 'Carnegie' stages of embryonic growth between 28 and 54 days [3]. Because the nascent nervous system is too delicate to show up in an MRI, individual nerves are not included in the segmentation process.

2.3.2 Modeling with Maya

The segmented vector outlines created in Fireworks were then imported into *Alias/Wavefront Maya*, a 3D modeling and animation package. These acted as reference guidelines for the creation of ten models showing the embryonic limb in each Carneige stage.

Because the developing nerves are too fine to image using conventional MRI, the nerves were added manually as sub-division surface cylinders. To ensure accuracy, reference was made to photographs of the brachial plexus of a dissected rat embryo which had been enhanced with dye injection and nerve staining [1]. These were cross-referenced with diagrams of a human brachial plexus and models constructed from an online version of the visible human dataset (*figure 3*) [12].

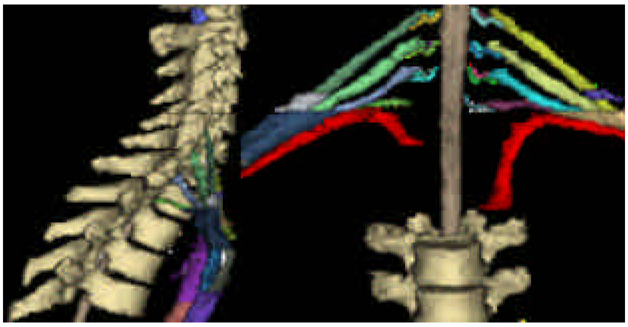


Figure 3. Sagittal and coronal views of an adult human brachial plexus.

2.3.3 Maya's advanced toolset

Maya is the market leader for computer graphics imagery within the film industry. It is the visual backbone of numerous movies from *Lord of the Rings* to *Finding Nemo* and has a powerful set of tools for Polygonal, NURBS and sub-division surface modeling [15]. Functionality includes editors for general, character and non-linear animation, forward and inverse kinematics, skinning, rigid-body dynamics, soft-body dynamics, particle effects, cloth, hair, fur, liquid simulation and integration with film footage [8]. A decade ago Maya required high-end Silicon Graphics computers. Now a standard Desktop PC with one or two Gigabytes of RAM and a good graphics card is adequate to run the software.

2.3.4 Advantages of Maya

- The most advanced modeling and animation tools commercially available.
- Highly customizable using the C++ API or the MEL scripting language.
- Previously prohibitively expensive, now a large educational discount brings it within reach of more modest budgets (reduced from £5000 to £400).
- The 'Personal Learning Edition' (PLE) provides all the functionality of the main product for free and is invaluable for training purposes (the PLE license forbids commercial use and renders all output with copyright watermarks).
- High quality 'master' datasets can be created and then outputted in a range of formats, for example VRML, FLASH, SVG, QuickTime or AVI.

2.3.5 Disadvantages of Maya

- Steep learning curve.
- Complex interface.
- Both artistic and technical skills required.
- Requires 3rd party software such as *VTK* to convert volume data sets into surface datasets.

3. CONTENT

3.1 Relevance

Medical students require information about the structure of the body but this propositional knowledge needs to be learned within an appropriate educational context. Models in themselves are of little educational or professional value without a carefully considered framework for delivery. Validation of 3D courseware in the WebSET project [11] demonstrated that supplemental material in the form of lecture notes, photos, and videos enhances the capacity for students to learn from 3D simulations (see section 4.3.3).

The level of detail for the brachial plexus project was tailored to suit a medical student learning the developing interconnections between nerves, muscle and bone for the first time. MRIs and diagrams from books were used as a reference to ensure accuracy but the 3D models themselves were stylised and exaggerated. For example, the layout of the brachial plexus was topographically accurate but the size of the nerves was shown larger than in reality in order to clarify the interconnections between structures.

3.2 Translating existing content

Ten 3D models were rendered as high quality static images to illustrate a storyboard of key moments from the original lecture.

The storyboard was a working document, refined several times by the project team. Once finalised, a hard copy was printed as an A3 poster which was used as a blueprint for creating the finished learning resource.

4. FUTURE WORK

4.1 Testing and evaluation

The first prototype of the storyboard incorporating the brachial plexus models will be presented to students who will be asked to use it as part of their support material and to complete a questionnaire evaluating its usefulness as a supplementary learning tool.

The finished version will be evaluated using two groups of students. The first will use the learning resource, the second will follow conventional methods of learning anatomy. Both groups will undertake an assessment to determine the effects of using the tool.

4.2 Reuse of models for postgraduate study

We plan to reuse the developed brachial plexus models to demonstrate effects of disease and injury upon the Plexus (brachial plexopathy).

4.3 Means of delivery

The primary methods of delivery will be an undergraduate intranet website and a CD-ROM. In addition we will experiment with a variety of other formats to assess more novel ways of delivering the learning resource.

4.3.1 Intranet site using Flash and Java Plug-ins

Maya has export filters that allow polygon, NURBS or sub-division models to be exported as a VRML file which can subsequently be embedded into a standard Java applet. This will allow students to manipulate the model of the brachial plexus in real time and control the speed of growth. Flash will be used where a 2D illustration is more educationally useful. Maya facilitates the creation of Flash content with a vector renderer which exports animations as a Scalable Vector Graphics (*.SVG) file that can easily be converted into Flash (*.FLA) format.

4.3.2 Java anatomy library within a Virtual Learning Environment (VLE)

VLEs such as WebCT or Blackboard are now commonplace in medical education both in the UK and abroad. VLEs bring several tools together in one password-protected online environment. Typical functionality includes a discussion board, multiple choice generators, downloadable lecture materials, chat room, whiteboard, student tracking and image databases. A common frustration for teachers is the lack of tools for subject specific teaching. One solution to the problem is to use the VLE as a password-protected shell from which to load additional applications in a new window. The brachial plexus will be the first exemplar of a searchable library of anatomical structures.

4.3.3 WebSET

WebSET (Web-based Standard Educational Tools) is a European Union funded project to develop web-based educational courseware [14] [4]. It comprises a set of tools which allows 3D simulations to be integrated into Web-based training packages using a standard Java plug-in (the Cortona VRML Client). WebSET has been used to simulate surgical procedures such as a lumbar puncture (using a needle to draw fluid from around the spinal cord), allowing a student to practise the technique within a collaborative environment monitored by a teacher. Lecture notes, videos, MCQs and performance metrics are used to supplement the 3D simulation. WebSET has been evaluated and shown to place students higher up the learning curve towards surgical competence before performing a real procedure [11].

WebSET will allow students to collaboratively explore the brachial plexus. This will be of particular interest to postgraduate students studying the pathology of the brachial plexus under the supervision of a lecturer.

4.3.4 Web services and online gaming

The Joint Information Systems Committee (JISC) in the UK is funding a three year Technical Framework Programme to develop a set of educational web services with open specifications and standards [5]. It aims to develop common service definitions, data models, and protocols to allow commercial, homegrown, and open source components to work together. In practice this means that the next generation of VLEs will move away from the monolithic browser / server model towards modular ‘Personal Learning Environments’ running as stand-alone Java or C++ applications on a client’s computer. If successful this approach will permit a new generation of sophisticated graphics engines to run without the limitations of web browsers, VRML and Java applets. The graphical quality of games such as the MMORPG Star Wars Galaxies is a good indication of how collaborative online learning environments may look in several years time. Much work needs to be done in assessing which elements of collaborative 3D game environments could be usefully adapted for educational purposes.

As an intermediate step, we intend to port the brachial plexus model into a range of web-enabled open-source game engines to experiment with current games technology. The forthcoming Half Life 2 is particularly suitable as it has an open SDK, a sophisticated graphics editor and is compatible with Maya [6].

4.3.5 Immersive VR

Imperial College London has a number of immersive VR facilities ranging from a full size VR suite running on an SG Reality Monster to an Elumens VS3 Dome. We will investigate the effectiveness of group delivery of the Brachial Plexus learning resource through these immersive VR facilities in comparison with the other proposed means of delivery.

4.3.6 Surgical simulation with haptics

The process of hardening and preserving a cadaver alters the colour, texture and feel of human tissue. Developments in simulation technology may allow future models to convey the touch sensations of virtual dissections using a haptic interface.

5. CONCLUSIONS

5.1 Conclusions to the brachial plexus project

Although the learning resource is yet to be finished and evaluated, initial indications are that it will be a useful supplement rather than a replacement for traditional methods of learning anatomy.

Maya is a powerful modeling tool that has yet to be fully exploited outside the film/animation community. Even though it is labour-intensive and a degree of trial and error is necessary to get good results, its functionality and export capabilities make it a good alternative when the more standard modeling route is not possible, for example due to inadequate results from imaging data.

The process of refining a storyboard amongst a project team is a working strategy that helps ensure proper integration of the 3D technology into established curricula. It also provides an external visual reference that helps to bridge the gap between staff from different educational, technical and medical specialties.

5.2 General conclusions

The introduction of interactive collaborative environments such as WebSET is a logical development beyond static webpages embedded with VRML and Java plug-ins. However, the graphical limitations of web browsers remain. If the Technical Framework Project proposed by JISC in the UK is successful, modular web enabled graphics engines running on a standalone PC may replace the browser altogether. As graphical fidelity increases, we will also need to determine what level of graphical and haptic realism is educationally useful for anatomy teaching.

6. ACKNOWLEDGMENTS

Our thanks to Dr Peter Cashman at Imperial College London for allowing the use of the knee segmentation images.

7. REFERENCES

[1] Aizawa Y, Isogai S, Izumiyama M, Horiguchi M. “Morphogenesis of the primary arterial trunks of the forelimb in the rat embryos: the trunks originate from the lateral surface of the dorsal aorta independently of the intersegmental arteries”. Anat Embryol (Berl). 1999;200(6):573-84.

[2] Bartz D, Mayer D, Fischer J, Ley S, Del Rio A, Thust S, Heussel C, Kauczor H.-U, Strasser W. “Hybrid segmentation and exploration of the human lungs”. Proceedings of IEEE Visualization 2003.

[3] Carnegie stages: http://embryo.soad.umich.edu/carnStages/carnStages.html (accessed 15.08.2004).

[4] Dodd A, Riding M, John N.W., “Building Realistic Virtual Environments using Java and VRML”, Third Irish Workshop on Computer Graphics, 2002, Dublin, Ireland: 53-61.

[5] JISC Technical Framework: http://www.jisc.ac.uk/index.cfm?name=elearning_framework (accessed 15.08.2004).

[6] Half Life 2 website: www.halflife2.net/ (accessed 15.08.2004).

[7] Lorensen W. E., Cline H. E.: “Marching cubes: A high resolution 3D surface construction algorithm”. Proceedings of the 14th annual conference on Computer graphics and interactive techniques 1987: 163-9.

[8] Maya functionality: www.alias.com/eng/products-services/maya/file/maya6_features_in_detail.pdf (accessed 15.08.2004).

[9] McLachlan JC, Bligh J, Bradley P, Searle J. “Teaching anatomy without cadavers”. Medical Education 2004;38: 418-24.

[10] Mitchell, Barry S & Stephens, Christopher R. “Teaching anatomy as a multimedia experience”. Medical Education 2004;38(8): 911-912.

[11] Moorthy K, Mansoori M, Bello F, Hance J, Undre S, Munz Y & Darzi A. “Evaluation of the benefit of VR simulation in a multi-media web-based educational tool”. Proceedings of Medicine Meets Virtual Reality 2004, IOS Press, January 14-17 2004, Newport Beach: 247-252.

[12] Visible human website: http://visiblehuman.epfl.ch/ (accessed 15.08.2004).

[13] Visualisation toolkit website: http://public.kitware.com/VTK/ (accessed 15.08.2004).

[14] WebSET website: www.hoise.com/vmwc/projects/webset/articles/websetHome.html (accessed 15.08.2004).

[15] Warren, Joseph D. Subdivision methods for geometric design : a constructive approach. San Francisco; London: Morgan Kaufmann, 2002.

A virtual reality user interface for learning in 3D environments

George Katsionis
Department of informatics
University of Piraeus
80 Karaoli & Dimitriou Str. Piraeus
18534, Greece
+3-210-5813659
gkatsion@kman.gr

Maria Virvou
Department of informatics
University of Piraeus
80 Karaoli & Dimitriou Str. Piraeus
18534, Greece
+3-210-4142269
mvirvou@unipi.gr

ABSTRACT

Computer assisted learning is a common issue among the research communities globally. The benefits of such technologies are widely accepted, so more and more educational applications are being developed. However, educational software aims to be used by students for learning. Therefore educational applications have to be as attractive as possible to increase the engagement of students. To serve this purpose many researchers employ several aspects of multimedia technology to improve the aesthetics and the appeal of educational software. One very attractive and popular application area, that involves and demands advanced multimedia technologies and could be used for educational purposes, is computer games.

In this paper a virtual reality game for educational purposes is described. This tutoring system incorporates a virtual reality user interface similar to that of common commercial games. The game is enriched with student modeling mechanisms that ensure the individualisation of the interaction. In order to create the virtual environment of the game, Virtual Reality Modeling Language (VRML) was used. Implementation issues of the virtual reality environment, and associated matters are being discussed.

Categories and Subject Descriptors

K.3.1 [**Computers and Education**]: Computer Uses in Education – *Computer-assisted instruction (CAI), Computer-managed instruction (CMI), Distance learning.*

General Terms

Design, Experimentation, Languages.

Keywords

Educational software, Virtual reality, Student modeling, Computer games, Intelligent Tutoring Systems.

1. INTRODUCTION

Children and adolescents are often fascinated by electronic games. Indeed, it has been widely acknowledged that electronic games are part of the popular culture of many children [18]. For example, Papert [16] acknowledges the fact that the greatest amount of children's time with the computer is devoted to playing games and Griffiths and Hunt [10] who conducted a study among adolescents, found that approximately one third of the subjects of their sample played computer games every day and the same amount played once a month.

Although children's fixation with these games initially alarmed parents and educators, educational researchers soon questioned whether the motivation to play could be tapped and harnessed for educational purposes [15]. Hence a lot of researchers have recently highlighted the advantages of computer games relating to education. For example, Boyle [5] notes that games can produce engagement and delight in learning. In addition there are many researchers who encourage this entertaining aspect of education and have developed games for educational purposes ([3], [14], [19]).

Additionally, the scientific community in general has acknowledged the need for a high degree of adaptivity and dynamic individualisation to each student that educational applications of any kind should provide. To this end, Intelligent Tutoring Systems (ITSs) have made significant contributions. Indeed, as Self [20] points out, ITSs are computer-based learning systems, which attempt to adapt to the needs of learners and are therefore the only such systems, which "care" about learners in that sense. It is simple logic that response individualised to a particular student must be based on some information about that student; in ITSs this realisation led to student modelling, which became a core or even defining issue for the field [8].

For the purposes of this kind of research, in this paper, we present an Intelligent Tutoring System (ITS) that operates as a virtual reality game. This game is an educational application for teaching English orthography and grammatical rules, inside three-dimensional virtual worlds. The environment of the game aims at increasing students' motivation and engagement and is quite similar to that of common commercial games.

2. VIRTUAL REALITY GAME

The environment of a game plays a very important role for its popularity. Griffiths [11] after conducting a questionnaire and interview study, found that the machine's "aura" typified by characteristics such as music, lights, colors and noise was perceived as one of the machine's most exciting features for a large part of the population questioned.

Our educational application invites the culture of computer games for creating a language Intelligent Tutoring System that can be very engaging, motivating. In the case of language tutoring systems the use of computer games may additionally provide a cultural internationalisation and wide acceptance of these systems.

The environment of the virtual game is similar to that of most popular virtual reality games, which has many virtual theme worlds with castles, corridors and dragons that the player has to navigate through and achieve the goal of reaching the exit. The main similarity of this tutoring system with computer games lies in their use of a 3D-engine. However, this game unlike commercial computer games of this kind is not violent at all and is connected to an educational application. One must fight one's way through by using one's knowledge. However, to achieve this, the player has to obtain a good score, which is accumulated while the player navigates through the virtual world and answers questions concerning English spelling. These virtual worlds look like the one in Figures 1,2.

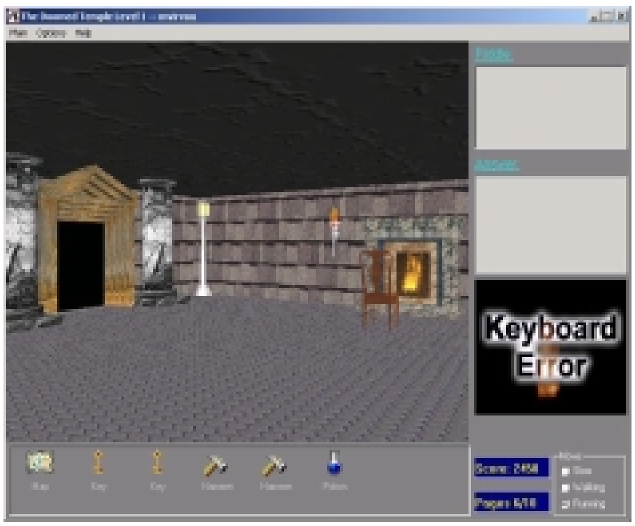


Figure 1. Virtual worlds of learning

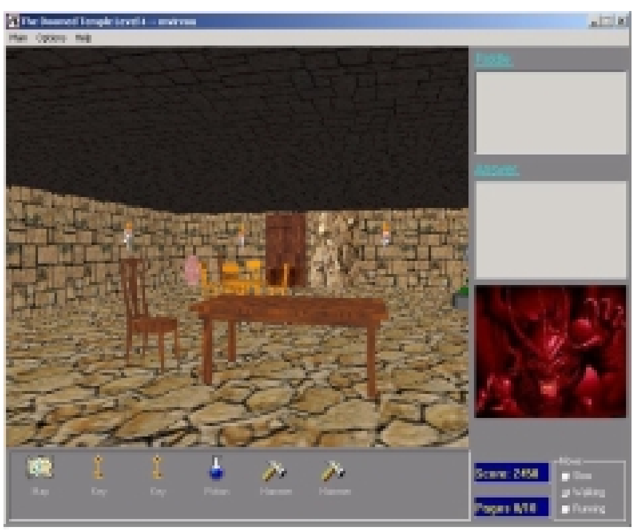


Figure 2. Virtual worlds of learning

As it can be seen in figures 1,2 part of the adventure of the game is the variety of inventory objects that the student can accumulate during the game and make the game even more attractive. There is map that every player has access to, and is an essential part of the game. The map shows an overview of the structure of the world, and is very useful for the navigation of the players in the world. The keys can be used to open doors that the students do not know the answer to the questions asked. Additionally there are other objects with useful functions including potions, hammers, teleports and books.

As we said the student-players of the virtual game have to answer certain questions in order to pass through the doors of the virtual worlds. When a student is asked a question s/he may type the answer in a dialog box. The ITS takes into account the history of answers of students and constructs a student-model for each one of them. The student characteristics that are being modelled concern the knowledge level of students (answers' results - errors) as well as some general actions. Additionally it provides individualised assistance, guidance, error diagnosis and navigation support based on these student models.

The game aims at helping students learn simple grammatical rules and check their spelling. For example, one set of grammatical rules concerns the plural form of nouns. Another one concerns the comparative and superlative form of adjectives consisting of one or two syllables. For both of the above sets, there are grammatical rules that categorise the nouns and adjectives into different categories. Thus, in the domain representation of the game, there are different categories of rules accordingly. While students interact with the game, it examines their answers in questions that relate to grammatical rules and categorises the errors that they may make. In this way, it may provide help according to the category of error.

The educational application communicates its messages to students through animated agents or through windows that display text. The user interface employs three types of animated agent: The dragon that is the virtual enemy of the player and is responsible for asking questions outside every door, the angel, which is the virtual advisor of the player and is responsible for providing hints and advice, and the virtual companion of the player who has the form of an elf.

The virtual advisor agent appears in some special situations. One of them is to inform the student to read new parts of the theory in order to continue playing the game, and be able to answer the questions that will follow. Another is to advice the student to repeat parts of the theory that s/he appears not to know well. Such information is obtained by making inferences about the mistakes of the student that are being tracked. In figure 3 below is illustrated an example of the advisor agent interacting with the student.

The virtual companion, who has the form of an elf, appears in cases where the student has given an answer, either correct or wrong, and the student has declined much from his usual actions (made a mistake in a category that s/he was always correct) or has made a repeated mistake. Then the virtual companion appears, asks the student about this situation and makes some notes. It simulates the behaviour of a friend for the student. However, it is essential not to become disturbing for the student by interrupting him/her continuously. The existence of the virtual companion has been considered quite important by many researchers for the purpose of improving the educational benefit of tutoring systems and promoting the student's sense of collaboration.

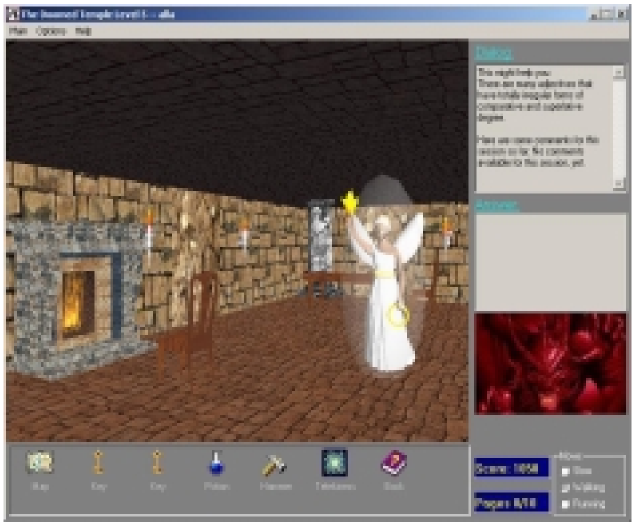


Figure 3. Virtual advisor agent

3. VIRTUAL REALITY ENVIRONMENT IMPLEMENTATION

Due to the advantages of computer games relating to education our main goal was to make an educational application that would have a user interface quite similar to the environment of common commercial games. The game category that we selected for our educational application was virtual reality games.

Virtual reality describes an environment that is simulated by a computer. Most virtual reality environments, including virtual reality games, are primarily visual experiences, displayed on a computer screen, and some include additional sensory information, such as sound through speakers. Because of using such advanced multimedia technologies, virtual reality gaming applications create rich and engaging environments. Such applications, which typically are very attractive and provoke a wealth of emotions to users, can become an advanced test bed for educational purposes.

There was no 3D commercial engine that we used to create our virtual application. We have created the 3D environment of the game from scratch. To create the virtual environment for our educational application we have used the Virtual Reality Modeling Language (VRML) [2]. VRML was originally designed to allow 3D worlds to be delivered over the World Wide Web. Using a VRML browser the user can explore this world, zooming in and out, moving around and interacting with the virtual environment. This allows fairly complex 3D graphics to be transmitted across networks without the very high bandwidth that would be necessary if the files were transmitted as standard graphic files. VRML can also include multimedia elements, such as texture images, video and sounds. The file extension for VRML files is ".wrl".

VRML was used to create the virtual worlds of the user interface of our educational application. However, due to luck of the VRML's language capabilities for creating a window application, all the rest parts of the functionality, the utilities, and the information kept for the students, were constructed with a visual programming language.

So the educational game's windows, forms, buttons, menus, etc, have all been created using DELPHI [12]. We then used a highly interactive 3D viewer to integrate our virtual worlds, which were created with VRML, into our DELPHI application. This 3D viewer is called Cortona VRML client [13] and it provides visual programming languages, DELPHI included, with an active-x component named Cortona in which virtual worlds created in VRML can be viewed by loading the according files. It should be mentioned here that the window application has the capability to use VRML as to make changes in the virtual worlds, create new .wrl files, and load them again. This implementation architecture is illustrated in figure 4 below.

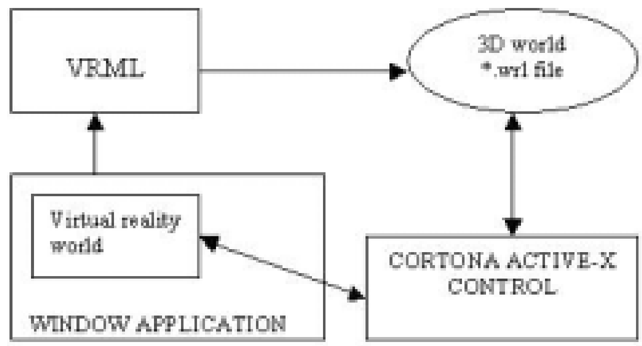


Figure 4. Virtual game's implementation architecture

The combination of a common and easy to use visual programming language to create the user interface of the educational application, and VRML for creating the virtual worlds to be embedded in it, provided the result presented in the above first three figures.

4. WEB3D PROBLEMS AND POTENTIAL SOLUTIONS

Educational applications express the need for a high degree of adaptivity and dynamic individualisation. Student modeling can cover this need. The shift from standalone to networked PC computing offers the capability of modelling a large population of individuals. Information about the learner is no longer stored locally on each learner's computer but in a central repository that can be accessed by any client of the application that requests it. This is usually implemented by using client-server architectures. A step forward for these architectures is educational applications that operate trough the Internet. Some researchers have used VRML language for creating simple educational applications that work through the Internet ([6], [22]).

At the beginning of our research our aim was to create a virtual game for educational purposes. An educational system that was a standalone, local application that worked as a virtual game was created. However, in order to exploit the benefits of computer assisted learning in multi-user environments and distance learning, we wanted to make that educational standalone-local application accessible to all students through the Internet. That meant that there should be a client-server program that would allow the students to interact with the application. Each student would play through the Internet, on the client installed on his or her system, and all of the information results would be stored in the server, located at the teacher's computer. In this way the teacher would always have access to all the information about the students' answers and their user-models.

In the past years there were some approaches to make client-server applications work through the Internet. For example, a method used for the deployment of an ITS over the web [1] was to take all the parts of the client program and provide them through the Internet. In that way, the client-server program worked by having the client as web pages on the Internet and the server located at the Web server where the web pages existed. The server was receiving information from the web pages.

A VRML world is made up of lots of simple shapes, such as cones and spheres, grouped together to form objects. The more shapes in the file, the more detailed the world, but at a cost of increasing the file size and the time taken by the browser to display the world. In our application there exist complex graphical objects that constitute of hundreds of simple shapes. Thus, it is made clear that a virtual reality game like our educational application, which has numerous objects inside its virtual worlds that are highly detailed, would take too much time to load in a web browser.

As a result, to take all the parts of the client program, including the VRML virtual worlds, and provide them through the Internet was out of the question. So what was best was to have the client installed at each student's computer, and by some way to communicate with the server at the teacher's computer through the Internet. For this goal we used the new trend in web communications, the Web services.

Web services can be defined as modular programs, generally independent and self-describing, that can be discovered and invoked across the Internet or an institution intranet [17]. Web services are interfaces that describe a collection of operations that are network-accessible through standardised XML messaging. Additionally Web services perform a specific task or a set of tasks. A Web service is described using a standard, formal XML notation, called its *service description*, which provides all of the details necessary to interact with the service, including message formats (that detail the operations), transport protocols, and location [9].

From a pure technology standpoint, Web services represent a shift toward the broad-based adoption of standard interfaces, like SOAP (Simple Object Access Protocol) [4], UDDI (Universal Description, Discovery and Integration) [21], and WSDL (Web Service Description Language) [7], which are used to access application functionality over the Internet.

More precisely, Web services are basically designed to allow loose coupling between client and server, and they do not require clients to use a specific platform or language. In other words Web services are language neutral. Mainly for these reasons among others, these services are becoming very popular. One important application area that may benefit from the advantages of this new technology is the field of education, where there may exist very demanding applications such as Intelligent Tutoring Systems (ITSs) that have to be transferred over the Web.

What is needed is to have some services and their interfaces mentioned on a Web server, for example the Internet Information Server (IIS) of the main computer of our computer lab. That makes these services known for every client program that can connect to the IIS of the main computer through the Internet. The services are implemented on the main computer in a server program that can be another executable application (EXE) or even a DLL (Dynamic Link Library). So a client calls a Web service form the IIS (Web server) by knowing its interface, and the IIS automatically sends the message for that service to the server program. The server program also aware of the service's interface executes it.

In a similar way we have developed our virtual game and installed it in different computers, which calls Web services from our IIS server of our main computer for performing specific tasks. The interface language exploited to publish services and exchange messages is WSDL. DELPHI provides special components responsible to make this kind of connections by using interface languages like WSDL and SOAP. Our client (virtual game) is aware of the services that are implemented on the server program. So the VR-Game calls a Web service form the IIS, and the IIS automatically sends the message for that service to the server program (DLL) that executes it. One such task is the addition of a student's answer to the database, which is located at the main computer. Along with his answer there are stored its characteristics (was it correct, what kind of mistake). Other tasks are any questions regarding the information that exists in the database that keeps the students model. For example the companion agent frequently asks about the category of knowledge being taught, or the specific rule of this category, that the student makes most mistakes, as to provide help for the student. The resulting architecture of the system is illustrated in figure 5.

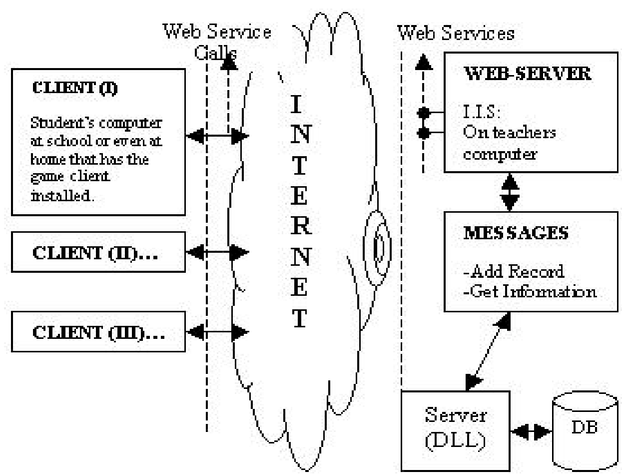


Figure 5**: The architecture.**

An example of the operation of the architecture of the system is the following. Every client, located in each student's computer can add an answer of that student to the database located at the main computer. So the client calls a Web service using an interface, through the Internet, from the Web server (Internet

Information Server in our example (IIS)) located at the main computer. Every call provides the Web service with a declaration and some variables, which contain the information about the student's answer. The Web server (IIS) then sends the message to the server program (.DLL in our example) that implements this interface. Then the server program executes the service and stores the answer to the database. In this way, from a user's point of view the game works in the same way as the standalone version.

Until now the results we are getting from our experiments show that the transfer to a Web-based ITS will help us a lot. We have not converted the whole program to work by using Web-services. It is certain that this is going to take us some time. Most of its' operations are executed directly from the client. We have implemented though some parts of it that are general for all the students and should be located at the teacher's computer. For example the database that keeps the user models and it is updated instantly from every client. The main usability difference that we have gained is that through Web services we are managing to make this demanding application to work through the Internet.

5. CONCLUSIONS

Educational applications may benefit from the technology of virtual reality games, which can increase the students' engagement and motivation. Individualisation of such tutoring systems is achieved by monitoring users while they play and by creating and maintaining detailed student models. These models are used for adapting instruction and advice accordingly.

However, one major problem of this kind of educational application is the construction and use of the game itself. The virtual reality user interface of such an application is extremely demanding. In this paper we have examined an implementation technique of a virtual reality educational game, and have discussed the difficulties faced to make it work through the web.

6. REFERENCES

[1] Alpert S. R., Singley M.K. & Fairweather P.G. (1999) "Deploying Intelligent Tutors on the Web: An Architecture and an Example", *International Journal of Artificial Intelligence in Education*, 10, 183-197.

[2] Ames A., Nadeau D., Moreland J. "VRML 2.0 Sourcebook", *2nd Edition*.

[3] Amory, A., Naicker, K., Vincent, J. & Claudia, A. (1998). Computer Games as a Learning Resource. *Proceedings of ED-MEDIA, ED-TELECOM 98, World Conference on Education Multimedia and Educational Telecommunications*, Vol. 1, pp. 50-55.

[4] Box D., D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, D. Winer, "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/SOAP.

[5] Boyle T. "Design for Multimedia Learning". *Prentice Hall*, 1997.

[6] Chover M., Ó. Belmonte, I. Remolar, R. Quirós, J. Ribelles. "Web-based Virtual Environments for Teaching", Proceedings of the Eurographics/ACM SIGGRAPH *Workshop on Computer Graphics Education*, 2002.

[7] Christensen E., F. Curbera, G. Meredith, S. Weerawarana (eds.) "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, "http://www.w3.org/2001/NOTE-wsdl-20010315".

[8] Cumming G. & McDougall A.: "Mainstreaming AIED into Education?", *International Journal of Artificial Intelligence in Education*, Vol. 11, (2000), 197-207.

[9] Gottschalk K., Graham S., Kreger H. & Snell J. "Introduction to Web services architecture". *IBM SYSTEMS JOURNAL*, VOL 41, NO 2, 2002.

[10] Griffiths, M. D. and Hunt, N. (1995) Computer game playing in adlescence: Prevalence and demographic indicators. *Journal of Community and Applied Social Psychology*, 5, pp. 189-193.

[11] Griffiths, M.D. "Technological addictions". *Clinical Psychology Forum*, 76, 1995, pp. 14-19.

[12] http://www.borland.com/

[13] http://www.parallelgraphics.com

[14] Inkpen, K., Upitis, R., Klawe, M., Lawry, J., Anderson, A., Mutindi, N., Sedighian, K., Leroux, S. & Hsu, D. (1994). "We Have Never-Forgetful Flowers In Our Garden: Girl's Responses to Electronic Games". *Journal of Computers in Math and Science Teaching*, 13(4), pp. 383-403.

[15] Mc Grenere, J.L. "Design: Educational Electronic Multi-Player Games. A Literature Review". *Technical Report* 96-12, the University of British Columbia, Canada, 1996.

[16] Papert, S. (1993). "The Children's Machine: Rethinking School in the Age of the Computers". *Basic Books*, New York, 1993.

[17] Pires P.F., Benevides M. & Mattoso M. (2002) "Building Reliable Web Services Compositions", Net.Object Days - WS-RSD'02, 551-562.

[18] Provenzo, E.F "Video Kids: Making sense of Nintendo". *MA: Harvard University Press*, Cambridge, 1991.

[19] Sedighian, K. (1997). "Challenge-Driven Learning: A Model for Children's Multimedia Mathematics Learning Environments". *Conference of Educational Multimedia & Hypermedia & Educational Telecommunications*, 1997, Calgary, Canada.

[20] Self J. "The Defining Characteristics of Intelligent Tutoring Systems Research: ITSs Care, Precisely", *International Journal of Artificial Intelligence in Education* ,10 , 1999, pp. 350-364.

[21] Universal Description, Discovery and Integration (UDDI) Version 2.0 Specification, June 2001, "http:/uddi.org/specification.html".

[22] Warrick, P.A. Funnell, W.R.J.:" A VRML-based anatomical visualisation tool for medical education", *Information Technology in Biomedicine*, Vol. 2, (1998), 55-61.