Advances in X3D multi-user virtual environments

Ch. Bouras
Computer Engineering and Informatics Dept. (CEID), Univ. of Patras, Greece, and Research Academic Computer Technology Institute (CTI), Greece
bouras@cti.gr

A. Panagopoulos
Computer Engineering and Informatics Dept. (CEID), Univ. of Patras, Greece
apanagop@ceid.upatras.gr

Th. Tsiatsos
Research Academic Computer Technology Institute (CTI), Greece
tsiatsos@cti.gr

Abstract

This paper presents the current state of the art in web3d multi-user technologies and more specifically networked virtual environment platforms that support the X3D standard, which are currently limited in number, and also the existing X3D browsers that enable these applications. Furthermore, this paper presents the architecture of a platform, called EVE, which was developed to support multi-user virtual environments and was initially based on VRML, as well as the implementation regarding the process of upgrading this platform in order to fully support the X3D standard, using the open-source Xj3D library. Finally, the benefits of shifting to the X3D standard are presented, and some of the problems that currently exist are discussed.

1. Introduction

The maturation of the Internet and the need for electronic communication formed the basis for the research and development of collaborative telecommunication applications. Multi-user 3d virtual environments are a powerful form of this type of applications, enabling the users to share a common 3d space and interact with each other as well as with the environment surrounding them, in order to collaboratively solve problems or aid learning processes.

The market as well as the research on collaborative telecommunication applications has been continuously growing during the last years. Two key features mainly characterize these applications. The first one is to enable co-ordination of activities without the need for participants to move physically. The second one is to share information between participants. The information shared in collaborative applications can be as various as document, 2D audio-visual natural representations of participants, 3D synthetic objects, or complex virtual environments.

Till now, the most important applications have been audio/graphic and videoconferencing, but it is expected that new technology and network development will generalize multimedia information on the Internet as well as collaborative systems, where groups of users can act and interact around this information. These key technical factors are very likely to allow the improvement and extension of the current application offering.

This expectation is supported by the current development inside the information technology world. Indeed, the networks capable of supporting the applications in order to transmit complex information, such as interactive multimedia data, are constantly increasing (e.g. xDSL). Moreover, the Internet and the Web, exported to these environments, bring new paradigms, techniques and opportunity for collaborative applications.

Networked Virtual Environments (NVEs) or multi-user virtual environments are very suitable for supporting a wide range of collaborative applications [8]. Networked Virtual Environments are multi-user virtual worlds, namely computer generated spaces, where participants represented by avatars can meet and interact. Nowadays, the use of NVEs is one of the most promising uses of virtual reality. Using NVEs as communication media, we can offer to members of virtual communities the advantage of creating proximity and social presence, thereby making participants aware of the communication and interaction processes with others [2]. Furthermore, NVEs can support collaborative applications by offering: (a) high level of presence of the users, (b) multi-modal

user-to-user interaction via chat, voice communication and gestures, (c) easy and friendly user-system interaction, and (d) scalability.

On the technological field VRML has been replaced by the new X3D standard for delivering 3D graphics over the web. X3D is an Open Standards XML-enabled 3D file format designed to enable real-time communication of 3D data across all applications and network applications. It has a rich set of features for use in engineering and scientific visualization, CAD and Architecture, Medical visualization, Training and simulation, multimedia, entertainment, educational, and more. X3D is a considerably more mature refined standard than VRML and obviously a better development choice than continuing to use VRML for the following reasons [1]:

- It is VRML compatible.
- It uses XML encoding to integrate smoothly with other applications.
- X3D scenes and environments operate predictably between different browsers/players.
- It is componentized, which means that it allows for the specification of profiles tailored to a particular large market segment (e.g., CAD, Medical, Visualization).
- X3D authoring for any player is consistent and simpler.
- It is more feature-rich than VRML by integrating a large number of features requested for VRML.
- X3D applications can be certified as reliable and predictable.
- An X3D open source conformant application (called Xj3D) is available as a developer resource.
- X3D binary format offers encryption (i.e. security) and compression (i.e. speed).

All the above reasons on the one hand make attractable the usage of X3D standard and on the other hand they suggest to migrate from VRML to X3D.

However this migration is in its early stages concerning networked virtual environments. In other words there are not many multi-user virtual environments platforms that are supporting X3D standard. This situation could be not only an obstacle on developing new collaborative 3D applications that exploit X3D's advantages but also a problem for X3D's spread.

This paper presents the current state of the art on X3D enabled networked virtual environment platforms. Furthermore, this paper presents the main implementation issues of developing such a platform by describing EVE X3D compliant platform.

This paper is structured as follows: The next section presents the state of the art on X3D enabled networked virtual environment platforms and browsers. Following this, the implementation issues, concerning the development of an X3D enabled NVE platform, are presented in section 3. Finally, some concluding remarks and planned next steps are briefly described.

2. State of the art

This section presents the state of the art on X3D enabled networked virtual environment platforms and browsers. Generally speaking there are many networked virtual environment platforms either commercial products or research platforms. The most significant commercial networked virtual environment platforms are the following:

- blaxxun platform (www.blaxxun.com)
- Bitmanagement solution (www.bitmanagement.de)
- Active Worlds (www.activeworlds.com)
- Octaga (www.octaga.com)
- Sense8 (sense8.sierraweb.net/)
- ParallelGraphics (www.parallelgraphics.com)
- SmartVR's SmartVerse (www.smartvr.com/)

The most significant research platforms are the following:

- DIVE: Distributed Interactive Virtual Environments (http://www.sics.se/dive/) [4]
- SPLINE: Scalable Platform for Large Interactive Environments (http://www.merl.com/projects/spline/)
- VLNET: Virtual Life Network [5], [6]
- SmallTool [3]

However, some of the above platforms are not supporting X3D standard at all and some of them are supporting X3D standard partially.

Furthermore, there are some X3D browsers (plug-ins, applets, etc.) that could be used for accessing X3D content in conjunction with a networked virtual environments server in order to support X3D networked virtual environments. The following paragraphs present these platforms and X3D browsers.

2.1 X3D Enabled NVE Platforms

The most significant solutions concerning X3D compliant NVE platforms are:

- **Blaxxun Platform 7 along with blaxxun3D Live 2.2**. Blaxxun Platform 7 is a server based commercial platform for creating and deploying 3D collaborative applications such as virtual classrooms, enterprise web conferencing, etc. This platform can support X3D multi-user virtual environments by deploying the multi-user version of the blaxxun3D browser (applet). Blaxxun3D is a java applet, which allows the user to access 3D environments without installation. It supports simple X3D profiles and can be programmed through its JavaScript or Java interface. Both the server and the blaxxun3D client are commercial products.

- **Bitmanagement solution.** Another solution for accessing X3D multi-user worlds is by using Bitmanagement's VRML/X3D browser (BS Contact) along with VR4ALL (http://www.vr4all.net/) server. BS Contact player supports X3D/XML, encrypted content, H-Anim/NURBS and Contact style MultiTexturing nodes, HTML and Flash composition in 3D scene as well as other extensions are supported. It is a commercial product which can run on Windows platforms.
- **Octaga**. Octaga is a commercial integrated suite of 3D players, server and development environment (SDK) for supporting, among others, collaborative virtual environments for education, and distributed simulation. Octaga Professional client is a commercial standards-compliant viewer for 3D and multimedia, which supports the whole profile of X3D and MPEG4. The Octaga Professional has multi-user functionality enabling conferencing and collaboration between users connected to the Octagon Server. Octaga server is a multi-user MPEG-4 based server allowing many users to collaborate in a common virtual environment.

Table 1 presents the above X3D compatible networked virtual environment platforms in more detail.

2.2 X3D browsers

The most significant X3D players (that are not part of an integrated NVE platform) are the following:

- **Flux Player** (www.mediamachines.com). Flux Player is fully compliant X3D commercial player for Windows. Flux is an ActiveX control that loads X3D and VRML97 scenes, provides user-level navigation features and connects to other web page elements via XML/DOM scripting.
- **FreeWRL** (freewrl.sourceforge.net). It is a X3D/VRML open-source browser and plug-in. There are source and binary distributions of FreeWRL available under GPL-style license. It supports javascript, EAI (External Authoring Interface), and java .class invocation and it can run over MacOS X, Linux, Unix, IRIX and Java compatible platforms.
- **OpenWorlds Horizon** (www.openworlds.com). OpenWorlds is a free, customizable, extensible browser, which offers very good rendering quality and it supports VRML and X3D as well as extensions for real-time shadows, reflection mapping, true reflections (mirrors), and NURBS. OpenWorlds browser runs over windows or SGI Irix and it could be extended by through the Horizon that should be purchased.
- **Xj3D** (www.xj3d.org). Xj3D is the result of a project of the Web3D Consortium focused on creating a toolkit for VRML97 and X3D content written completely in Java. The toolkit contains a stand-alone viewer and can be used either for importing VRML content into custom applications, or for the creation of a fully-fledged browser. Xj3D toolkit is free under GPL-style and supports Windows, MacOS X, Linux, Unix, IRIX and Java platforms.

Platform/ Features	Blaxxun	Bitmanagement	Octaga
X3D support	Partially: (a) X3D Interactive Profile support (b) Complete VRML97 ISO standard support	Partially: (a) X3D XML Encoding of VRML Scenes (b) H-Anim/ NURBS (c) Contact style Multi-Texturing (d) No additional X3D nodes are supported. (e)Very good rendering quality	Partially: (a) Core profile support, extra nodes and new nodes in recent amendments. (b) Partial support for the NURBS component. (c) MPEG-4 Interactive profile support (d) DISP, H-Anim, Geospatial components are not supported (e) Good rendering quality
Type of architecture	Client – Server - Client: blaxxun3, JavaApplet - Server: blaxxun Platform 7	Client – Server - Client: BS Contact VRML/X3D, BS SDK Package - Server: VR4ALL	Client – Server - Client: Octaga Professional - Server: Octaga Server
H-anim support	Yes	Yes	No
Streaming video support	Yes	Yes	In progress
Voice over ip support	Yes	Yes	No
Extensibility	Yes	Yes	Yes
Scripts support	Yes (VRMLScript / JavaScript)	Yes	Yes (Java Script/ ECMA Script.)
3D authoring tools	bxx AvatarStudio, bxx Placebuilder, 3D StudioMax, blender3d, etc.	BS Exporter BS Internet Optimizer	API for custom node development
Operating system	Server: Windows (NT, 2000, XP), Linux Client: blaxxun3D JavaApplet (all)	Windows (98, NT, 2000, Millenium, XP)	Win32 and Linux
Financial info	Commercial	Commercial	Commercial

Table 1: X3D compatible networked virtual environments platforms

2.3 Results

According to the above paragraphs, there are few X3D enabled NVE platforms today available. Concerning the X3D browsers there are more solutions. Having in mind the support of collaborative virtual environments there are two main categories of solutions:

- The exploitation of a ready-made X3D enabled NVE platform
- The integration of an X3D compatible player in a current (VRML) NVE platform

Concerning the first category, the most promising X3D enabled NVE platforms today are Blaxxun, Bitmanagement and Octaga solutions. Almost all these platforms partially support X3D standard, and offer good rendering functionality. However, all these solutions are commercial and any extension and or programming, that requires additional technical implementation, cost additionally due to the additional cost of the respective SDKs. Furthermore, a commercial solution may have the risk of a closed solution due to each company's extensions to the standards.

Concerning the second category, two attempts are in progress today:

- The development of Flux multi-user server along with X3D Flux client: This effort is in progress by Media Machines (http://www.mediamachines.com/). It will be based on Flux server and client supporting the latest X3D versions (i.e. immersive profile of X3D), NURBS and H-Anim with single-mesh (skinned) characters. This platform will run on Windows, and later on Macintosh and Linux. It should be noted that this platform will be a commercial product.
- The integration of FreeWRL client with Multicast Virtual Interchange Protocol (MVIP-II, http://sourceforge.net/projects/mvip/): More specifically the MVIP-II project uses FreeWRL in VRML mode, based on a multicast peer-to-peer architecture. FreeWRL supports partially the X3D standard. It also supports H-anim, VRML EAI/scripting, and X3D parsing. The MVIP-II component will run on Linux and FreeWRL component runs on Unix/ Mac OSX platforms. The result of this project will be an open source platform.

According to the above, it is obvious that in order to support X3D collaborative virtual environments the most mature solutions are the commercial platforms. However, the cost in this case is high.

The authors of this paper have decided to develop an X3D enabled NVE platform by integrating an X3D compatible player with a current (VRML) NVE platform. The platform chosen called EVE (http://ouranos.ceid.upatras.gr/vr, [7]). The X3D player chosen called Xj3D toolkit.

3. Implementing a NVE platform using Xj3D

As described before, our implementation of a networked virtual environment platform using X3D was based on a previous platform developed by the authors, called EVE. This platform used the VRML97 standard and offered easy and transparent sharing of virtual environments, where multiple users could navigate, communicate and interact with each other and the environment. Our goal was to upgrade this platform in order to fully support the X3D standard and take advantage of its additional functionality.

The EVE platform is based on a client-multiserver architecture, which allows a simple sharing of the computational load among multiple servers. The main servers used by the platform are the connection server, the VRML server (or data server) and a series of application servers, which add specific functionality such as audio and text chat to the platform. The architecture of the EVE platform is displayed in Figure 1.

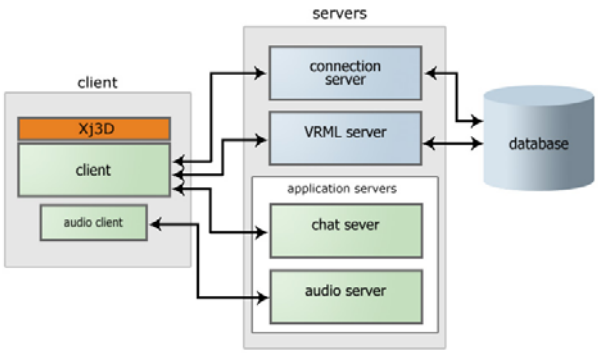

Figure 1: Architecture of EVE

Modifying an NVE platform to support the X3D standard includes modifications both on the server and the client side. These changes and the rationale behind them will be discussed in more detail in the next paragraphs.

3.1 Server modifications

EVE, as already mentioned, consists of several servers. The only server that needs to be changed to accommodate for the X3D standard is the VRML server or data server. This server has several functional modules. Most of them need not be changed. The duties of the VRML server include parsing VRML code to create new objects or worlds, storing and updating a local copy of the virtual universes served by that server, executing scripts when

they are activated by sensors or user actions, and also sending updates to the users and handling user movement and interaction.

The first and most obvious change necessary is to create new internal structures to store and manipulate the new nodes defined in the X3D specification and the hierarchy of an X3D world. We added a set of classes, one for each node defined in the X3D standard, which where semi-automatically generated by the definitions of the nodes, and included all functionality needed by the server to manipulate those nodes. Furthermore, there were some minor changes on the classes that represent the fields of the nodes that make up the virtual world, and some classes where added to provide a Scene Access Interface (SAI)-compatible interface for the scripts that are executed on the server side.

Furthermore, a new parser was created, based on Xerxes (http://xml.apache.org/), which parses X3D code in XML form, in order to create the internal representation of nodes and the node graph. Also, the existing VRML parser was extended in order to parse XML code in VRML form (.x3dv files).

Due to the close relationship between the VRML and the X3D standards, the changes that were necessary in order to convert the data server to support X3D were minor – with the exclusion of the addition of the XML parser. On the client side, however, important decisions had to be made, and a major redesign was forced.

3.2 Client modifications

The original EVE client was a java applet, which handled all communication with the servers, and used the EAI (External Authoring Interface) to communicate with the VRML browser and manipulate the virtual environment displayed to the user. The java applet did not do the rendering of the virtual world or the interaction with the user – an external VRML browser was chosen instead, to perform these tasks.

Stepping to an X3D solution, we were faced with two different ways to implement the client: Either use an X3D-compliant browser, such as BS Contact or Flux, or use a toolkit or open-source engine to implement a new client from scratch, which would handle all rendering and user interaction tasks. As discussed in previous paragraphs, most available X3D browsers either have severe functionality limitations, or are commercial. Using a commercial X3D browser is not acceptable for a platform such as EVE, which is intended to be free and open, and the limitations of the freely available X3D browsers made them inappropriate for our purpose.

As a result, the only way left to go was using a toolkit to create a client applet from scratch, which would handle rendering and user interaction, replacing the X3D browser. Some of the available are listed in the table below:

- Xj3D – A Java library that implements the necessary framework to create x3d browsers and applications that support VRML and X3D. It is based on Java3D and offers a satisfying feature set. Claims to support HAnim, NURBS, physics etc. However, it is under development, so some of these features are still buggy.
- X3Dtoolkit – A C++ library for loading/rendering X3D models. However, the feature set offered by the X3Dtoolkit is limited, and it is not sufficient to build a fully-featured X3D browser or an application with such capabilities.
- CyberX3D – A C++ and Java library for loading and displaying X3D worlds. CyberX3D supports only a limited set of the X3D nodes, aiming mainly at loading/storing X3D worlds and accessing the scene graph.
- Open ActiveWrl - Software development toolkit that allows the implementation of parallel immersive VRML/X3D browsers on heterogeneous cluster systems.

As can be seen by the above list, the most promising solution to create an X3D-enabled Java applet is Xj3D, which offers a flexible and cross-platform Java architecture and an almost complete feature set. Xj3D is discussed in greater detail in the next paragraph.

3.3 Xj3D

Xj3D is a project of the Web3D Consortium focused on creating a toolkit for VRML97 and X3D content written completely in Java.

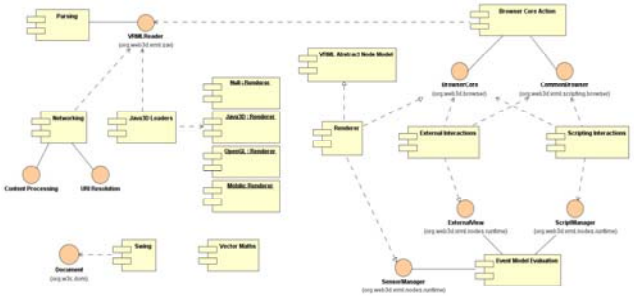

Figure 2: Xj3D general architecture

This toolkit may be used to import VRML content into a custom application, or to create a fully-fledged VRML/X3D browser. Development is financially supported through a number of institutes and companies. The initial impetus for this project was to create a file loader for the Java3D API and started with a grant of code from Sun Microsystems to the Web3D consortium. Over time, the project has grown in requirements and now encompasses many other features as well. For

example, it is being used as one of the main testing grounds to verify the work on the new X3D specification. It has moved beyond its original Java3D-only roots and now can work with many different renderers.

Xj3D is in continuous development. It has not reached version 1.0 yet, but the next release should be it. In the meantime, stable development snapshots are released, known as Milestones. The current release features installers for Windows, Linux and Mac OS X platforms. In addition, the following X3D components have been implemented: CADGeometry, DIS, GeoSpatial, H-Anim. Some custom extensions for Xj3D include: Rigid Body Physics, Particle Systems, Clipping planes, Picking Utilities, Abstract Device IO. New profiles include the CADInterchange profile. Elumens Dome support is also now standard. This release will be the last milestone release before 1.0.

The general architecture of the Xj3D is shown in Figure 2.

3.3.1. Implementing an X3D application with Xj3D. Xj3D offers all the necessary components to build a fully featured X3D application or X3D browser, using a flexible architecture. The virtual environment is represented by a Universe object, which holds a VRMLScene object with the node hierarchy. Several managers are used to handle every aspect of the functionality of the virtual world. There are managers for routes, sensors, script loading or HAnim hierarchies. Through the use of managers, an application can control most of the stages of presenting a scene to the user and receiving feedback from him/her. For example, various input devices can be supported by the InputDeviceManager and an application can have access to the culling and sorting of objects through the CullStage and SortStage objects. Loading of 3D objects is done using the appropriate parsers and loaders that are created by a parser factory. Xj3D uses the Java3D framework to load and render the 3D worlds. This way, OpenGL and a software renderer are both supported, and the efficiency and visual quality of the library are quite satisfying. Audio is supported as well, and there are also some extensions such as a rigid body physics engine, which is implemented using the well-known open source ODE engine.

3.3.2. Issues regarding Xj3D. While developing the X3D-compliant version of EVE, we came across several bugs and deficiencies of the Xj3D platform. The most important of them are the following:

- Problems with the H-Anim specification support: Xj3D has severe problems displaying H-Anim articulated humanoid models. These problems result in run-time exceptions, which make the browser thread die. We witnessed these problems not only when using our own test models and avatars, but also with the X3D example files provided by the Web3D Concortium. The lack of proper H-Anim support is an important problem when implementing an NVE platform, since it puts constraints to the creation of avatars. In fact, we finally had to drop the H-Anim support of Xj3D and resort to HAnim-like PROTOs and scripts that implemented the animation. Due to performance reasons, the use of smooth skinning is not feasible.
- Missing implementations for some nodes defined by the X3D standard, such the MovieTexture and Switch nodes. Regarding the MovieTexture node, while it was initially supported through the use of JMF, support was removed due to the numerous problems of the JMF implementation. This lack of support for MovieTexture nodes means that there is no way to have video easily integrated into a virtual world. However, video is an important means of communication and is widely used in many of the uses of NVEs, such as distance learning scenarios. MovieTexture will be supported in the next versions of Xj3D, probably using OpenAL for the implementation. Also, the Switch node is not correctly implemented in the current version of Xj3D, causing some problems, since it is a widely used and useful node.
- Problems with multitexturing and texture coordinate generation: There are still some problems when using multiple textures and automatic texture coordinate generation to achieve effects such as environment mapping.
- Strangely, Xj3D does not identify JavaScript scripts starting with the prefix “vrmlscript: “ or “ecmascript: “, although most VRML browsers do. That way, many VRML objects which have embedded JavaScript scripts, have to be converted so that the scripts start with the “javascript: “ prefix.
- An Xj3D-based browser is heavier regarding resources and more memory-intensive than some good implementations of VRML browsers such as BS Contact or Octaga.

In general, most of the bugs and problems of Xj3D are things we can live with, with the exception of HAnim and the lack of MovieTexture support, which are important and limiting the potential applications of the library. However, as Xj3D approaches version 1.0 (it will be the next stable release), we expect that these issues will be resolved.

3.4 Benefits

Implementing X3D support on both the server and the client side of the EVE platform was relatively easy, with relatively few issues that had to be resolved. However,

adding support for the X3D protocol offered significant benefits, improving the usability and flexibility of our platform and extending its feature set to include several new concepts of the X3D standard. These benefits can be summarized to the following:

- Better rendering quality, since X3D offers many additional useful features compared to VRML, that enable the creation of more realistic 3d worlds, with the most important being support for multitexturing, texture coordinate generation, NURBS etc.
- Much more stable and robust, compared to an EAI implementation
- Fully customizable, since the structure of the Xj3D toolkit offers great control by the program. Furthermore, custom extensions can be added to Xj3D, although this is a complicated task.
- Addition of meta-information to X3D nodes, using the meta tag, is very important in adding attributes which define the node's behaviour in a multi-user virtual environment and enabling a simpler, easier and more expressive way to implement different scenarios in the virtual world.
- The EVE platform now does not include any closed-source or commercial components at all, which is important in order to support the open-source philosophy.

As a result, the EVE platform currently supports the following characteristics:

- Flexibility and improved rendering quality, since the large set of all X3D nodes can be used to create worlds that are visually more appealing, as well as better defined compared to a VRML world.
- A consistent shared 3d virtual environment for all users connected to the platform, which is highly interactive and allows for all possible functionality of an X3D world to be reliably shared among all connected clients.
- An efficient physics system functioning locally on each client's machine, which is provided by the Xj3D library and based on the ODE open-source physics engine, as well as an efficient sound system.
- Previously existing functionality such as:
  - Text chat and audio communication, using H.323 for audio and chat bubbles for text chat.
  - User roles and user management, enabling a privileged user to expel other users, or use breakout rooms to assist the learning process in an e-learning scenario.
  - Support for avatar gestures and body language
  - A flexible client-multiserver architecture.

4. Conclusions- Future Work

This paper presents a state of the art of X3D enabled multi-user platforms as well as possible solutions for migrating from a VRML based multi-user platform to X3D. As seen in the previous chapters, there are some commercial platforms that offer satisfying support for networked virtual environments based on X3D. However, there is no ready research or free (open source) platform for the same purpose. A possible solution is the transformation of EVE research platform in order to support X3D multi-user virtual environments. As described in this paper, converting our existing EVE platform to support X3D was relatively easy and significantly enhanced the usability and efficiency of the platform. The X3D standard offers a fully featured set of nodes, enabling the world's author to create complex and visually appealing environments without having to deal with the restrictions of VRML, and the Xj3D library proved sufficient to replace the VRML browser we previously used. This replacement made our platform not only more robust but also much more efficient and flexible. Some issues regarding Xj3D still exist, mainly related to the missing implementations for some nodes defined by the X3D standard, but we hope that when version 1.0 is out, most of them will be solved. Our next step is the implementation of collaborative applications for supporting e-learning and co-operative work on top of EVE platform.

5. Acknowledgements

We would like to thank Tony Parisi (Media Machines), Peter Schickel (Bitmanagement Software GmbH), John Stewart (CRC Canada, FreeWrl), John Arthur (Octaga), Klaus Eberle (Blaxxun Technologies), for their valuable info concerning the state of the art on X3D enabled technologies.

6. References

[1] Web3D Consortium, http://www.web3d.org/

[2] C. Bouras, T. Tsiatsos, "Educational Virtual Environments: Design Rationale and Architecture", *International Journal of Multimedia Tools and Applications (MTAP)*, 2006 (to appear).

[3] W. Broll, "SmallTool - a toolkit for realizing shared virtual environments on the Internet", *Distributed Systems Engineering Journal, Special Issue on Distributed Virtual Environments, Vol. 5.* The British Computer Society, The Institution of Electrical Engineers and IOP Publishing Ltd, 1998, pp. 118-128.

[4] C. Carlsson, and O. Hagsand, "DIVE: a Multi User Virtual Reality System," *in Proc. of IEEE 1993 Virtual Reality Annual International Symposium*, VRAIS '93, Piscataway, NJ: IEEE Service Center, 1993, pp. 394- 400.

[5] I. Pandzic, T. Capin, N. Magnenat-Thalmann, and D. Thalman, "Towards Natural Communication in Networked Collaborative Virtual Environments", in Proc. of *FIVE '96*, Pisa, Italy, 1996.

[6] I. Pandzic, N. Magnenat-Thalmann, and D. Thalmann, "Realistic Avatars and Autonomous Virtual Humans in: VLNET Networked Virtual Environments", *Virtual Worlds in the Internet* (R. Earnshaw and J. Vince, eds) IEEE Computer Society Press, 1998.

[7] C. Bouras, A. Panagopoulos, N. Theoharis, T. Tsiatsos, “EVE - II: An Integrated Platform for Networked Virtual Environments”, *The Tenth International Conference on Distributed Multimedia Systems (DMS'2004)*, San Francisco, USA, September 8 - 10, 2004, pp. 81 - 85.

[8] C. Bouras, E. Giannaka, T. Tsiatsos, “Designing Virtual Spaces to Support Learning Communities and e - Collaboration", *5th IEEE International Conference on Advanced Learning Technologies*, Koahsiung, Taiwan, July 5 - 8 2005