Implementing Advanced Characteristics of X3D Collaborative Virtual Environments for supporting e-Learning: The case of EVE Platform

Christos Bouras[1, 2], Vasileios Triglianos[2], Thrasyvoulos Tsiatsos[3]
[1] Research Academic Computer Technology Institute, Greece
[2] Computer Engineering and Informatics Dept., Univ. of Patras, Greece
[3] Department of Informatics, Aristotle University of Thessaloniki, Greece
bouras@cti.gr, triglian@cti.gr, tsiatsos@csd.auth.gr

Abstract

Three dimensional Collaborative Virtual Environments are a powerful form of collaborative telecommunication applications, enabling the users to share a common three-dimensional space and interact with each other as well as with the environment surrounding them, in order to collaboratively solve problems or aid learning processes. Such an environment is “EVE Training Area tool” which is supported by “EVE platform”. This tool is a three-dimensional space where participants, represented by three-dimensional humanoid avatars, can use a variety of e-collaboration tools.

This paper presents advanced functionality that has been integrated on “EVE Training Area tool” in order to support: (a) multiple collaborative learning techniques (b) Spatial audio conferencing, which is targeted to support principle 3 (augmenting user’s representation and awareness). Furthermore the paper presents technological and implementation issues concerning the evolution of “EVE platform” in order to support this functionality.

Keywords

Collaborative Virtual Environments, X3D, Collaborative Spatial Design, Spatial Audio Conferencing, SIP.

Introduction

The maturation of the Internet and the need for electronic communication formed the basis for the research and development of collaborative applications. Collaborative Virtual Environments (CVE) is a promising form of this type of applications. CVEs might vary in their representational richness from three dimensional 3D graphical spaces, 2.5D and 2D environments to text-based environments (Snowdon et al. 2001). CVEs can enable 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.

Collaborative Virtual Environments are technologically based on Networked Virtual Environment (NVE) platforms. NVEs allow the communication and interaction of geographically separated users, inside 3D virtual worlds. This paper presents advanced NVE’s functionality that has been integrated on EVE platform (Bouras et al. 2001; Bouras & Tsiatsos, 2004; Bouras et al. 2005; Bouras et al. 2006) in order to support. More specifically, the main goal of this paper is to present the evolution of

EVE platform in order to support e-learning and e-collaboration scenarios in a more effective manner.

Since the early uses of collaborative virtual environments in learning, researchers have tried to establish a schema that incorporates some well known aspects, issues, elements and principles which should be taken into account during the design process of educational virtual worlds. The rationale behind the designers' decisions can have a significant effect on the appropriateness of the platform for education. Regarding the design adequacy of EVE for online learning purposes, we validated (as presented in the next section) the platform's features, philosophy and policies against the design principles presented in Bouras et al. (2008). These principles are the following:

- Principle 1 - Design to support multiple collaborative learning scenarios: A useful tool for collaboration would support the execution of many e-learning scenarios. E-learning scenarios can combine one or more instructional methods like role-playing, case studies, team projects, brainstorming, jigsaw and many more, as long as the environment supports their functional requirements.
- Principle 2 - Design to maximize the flexibility within a virtual space: Space parameters like size, architecture, facilities and the physical environment affect the way learners socialize (Koubek & Müller, 2002). In order foster educational value, virtual environments must fulfil the teacher's expectations for spatial and temporal flexibility. Therefore, due to the need for multiple functions within a collaborative online synchronous session, it should be possible to quickly reorganize the virtual place for a particular activity or scenario.
- Principle 3 - Augmenting user's representation and awareness: Combining gestures, mimics, user representation, voice and text chat communication, users can share their views and show others what they are talking about.
- Principle 4 - Design to reduce the amount of extraneous load of the users: The main objective of an e-learning environment is to support the learning process. Therefore, the users should be able to understand the operation of the learning environment and easily participate in the learning process.
- Principle 5 - Design a media-learning centric virtual space: The virtual space should be enhanced by multiple communication and media layers. Each media type (e.g. text, graphics, sound etc.) has its advantages. The virtual space should integrate many communication channels (e.g. gestures, voice and text chat etc.) in order to enhance awareness and communication among the users.
- Principle 6 - Ergonomic design of a virtual place accessible by a large audience: The designers of a virtual place should take into account that a virtual place for e-learning could be used by various individuals with different backgrounds and level of expertise in information and communication technologies.
- Principle 7 - Design an inclusive, open and user-centred virtual place: SL membership is free, anyone above 18 years old can join (there is also a separate world for teenagers) and the virtual content of the world is created by its users.
- Principle 8 - Design a place for many people with different roles: An e-learning system should support a variety of roles each with different access rights. For example, in a collaborative learning scenario the participants could be moderators, tutors, or learners. The virtual space should be designed accordingly in order to differentiate these roles.

The previous versions of EVE platform could support the majority of the above principles through the EVE Training Area tool (Bouras & Tsiatsos, 2002; Bouras et al., 2005). This tool is a three-dimensional space where participants, represented by

3D humanoid avatars, can use a variety of e-collaboration tools. However, the previous versions of EVE should be improved in order to support the following functionality:

- Multiple collaborative learning (CL) techniques (i.e. to support principle 1) and flexibility within a virtual space (i.e. to support principle 2).
- Spatial audio conferencing, which is targeted to support principle 3 (augmenting user's representation and awareness).

The improvements that were made and the new functionality that were added to the previous version of EVE platform are focused on the satisfaction of the above mentioned principles as well as on the enhancement of the EVE platform in terms of performance, compatibility and stability. The advancements presented in this paper concern: (a) full compatibility with the current version of Extensible 3D (X3D) standard (Web3D, 2009) and extension in order to offer X3D event sharing, even for dynamic created shared objects; (b) support for spatial audio conferencing; and (c) integration of a generic non-X3D events management;

This paper is structured as follows. In the next section EVE training area tool is presented and the way that every principle is met in this environment is described. In addition this section introduces the necessary functional as well as technological improvements in EVE platform which are needed for the support of the new functionality. The third section presents the related work done on X3D enabled NVE platforms; protocols for supporting 3D event sharing in NVEs; spatial audio conferencing; and collaborative design applications. The fourth section presents the integration of X3D event sharing mechanism in EVE platform. Afterwards, in fifth section, we present in detail the design of a spatial audio conferencing server and its integration in EVE platform. In the section that follows we describe a server dedicated to handle non-X3D events sharing. Finally, we present some concluding remarks and our vision for future work.

Supporting Collaborative Learning with EVE Platform: Affordances, Limitations, Functional and Technological Extensions

In the following paragraphs, EVE training area is presented and the way that every principle is met in this environment is described.

Furthermore, we are discussing the limitations of this tool concerning the support of:

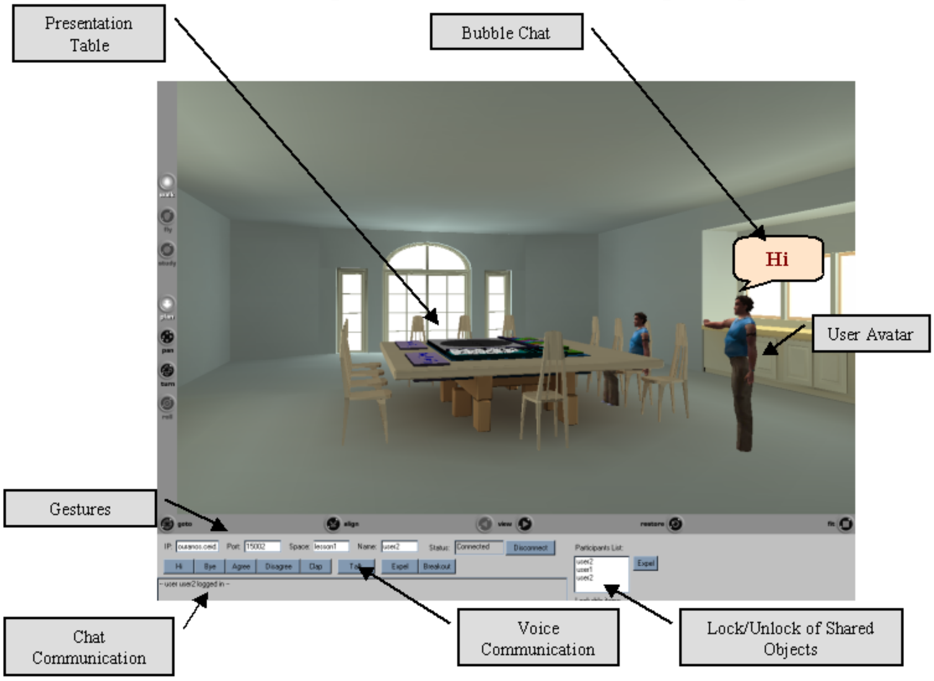
- Multiple collaborative learning techniques and flexibility within the virtual educational space
- Spatial audio conferencing

In addition, we are introducing the necessary functional as well as technological improvements in EVE platform which are needed for the support of the new functionality.

EVE Training Area

EVE Training Area is designed and implemented for hosting synchronous e-learning and e-collaboration sessions. As described in Bouras & Tsiatsos (2006), after the user-evaluation, the usability of this tool has been rated positive. It combines 2D and 3D features for providing the users with the necessary communication and collaboration capabilities. The main feature of EVE training area is the 3D representation of a multi-user virtual classroom. The user interface of the training area is depicted in Figure 1. The participants in the virtual classroom can have two different roles: tutor (only one participant) and students. In that way EVE training area meets principle 8.

The users that participate in the virtual classroom are represented by humanoid articulated avatars, which can support animations (such as walking and sitting down) and gestures for non-verbal interaction among the users. EVE's avatars support functions not only for representing a user but also for visualizing his/her actions to other participants in the virtual space, which also satisfies principle 3.



IP: ouranos.ceid Port: 15002 Space: lesson1 Name: user2 Status: Connected

-- user user2 logged in --

Figure 1: User interface of the training area

Available functions in EVE Training area are: Perception (the ability of a participant to see if anyone is around); Localization (the ability of a participant to see where the other person is located); Gestures (representation and visualization of others' actions and feelings. Examples are: "Hi", "Bye", "Agree", "Disagree", and "Applause"); Bubble chat (when a user sends a text message, a bubble containing the message appears over his/her avatar).

The virtual classroom is supported by various communication channels (principle 5) such as (a) audio chat, which is the main interaction channel, (b) 3D text/bubble chat, (c) non verbal communication using avatar gestures in order to provide a more realistic interaction among users, expressing, when needed, the emotion of each one to the others (Capin et al, 1999). Furthermore, EVE Training Area supports manipulation of users and shared objects by integrating two specific tools: (a) expel learner/participant and (b) lock / unlock objects. EVE Training Area integrates a "presentation table", which is the central point in the virtual space, in order to provide specific collaboration tools. Using the functionality of this table the users can present their slides and ideas, can comment on slides, upload and view learning material as well as view streaming video. The avatars of all participants in the virtual space can have a sit next to this table, viewing not only what is presented on the table but also the other participants. Furthermore, the user can change his/her viewpoint in order to zoom in and out on the presented material. The presentation table has the following functionality:

- 3D Whiteboard: The 3D whiteboard supports slide projection, line, circle and ellipsis drawing in a wide range of colors and text input in many sizes and colors. It also offers "undo last action" capability as well the cancellation of all previous actions on the whiteboard.

- Brainstorming Board: The brainstorming board can be used in a range of collaborative learning techniques for learners to present their ideas in a structured way. The users can create cards in three shapes (rectangle, circle and hexagon) and five colors attaching text on them. It should be mentioned that the shape and color of the cards is attached to a defined argument. They can also move and delete a card.
- Video presenter: Video presenter is used in order for the user to attend streaming video presentation/movies inside the 3D environment. The users have the capability to start and stop the movie. Supported formats are rm, mpeg, and avi.
- Library with drag and drop support: The users have the capability to drag and drop learning material on the table. This material is represented as a small icon on the backside of the table. When the user clicks on the icon the corresponding file is opened either on the whiteboard (if the corresponding file is picture or VRML object), on the video presenter (if the corresponding file is of rm, mpeg or avi type) or on a new pop-up window (if the corresponding file is not supported by the VRML format).

In order to augment the user's representation and awareness (and to satisfy principle 3), the usage of avatars along with gestures and additional icons attached to the avatar could be very helpful (Bouras & Tsiatsos, 2006). Examples of this functionality are the following:

- Bubble chat over the avatars head, which can be used in order to inform the participants of a session about the text chat input of this user. Figure 2a depicts the implementation of a bubble chat.
- User representation and avatar gestures for expressing actions and feelings. In Figure 2b, we can see an avatar of a user to visualize a "Hi" action by a gesture in the EVE training area (Figure 2b).

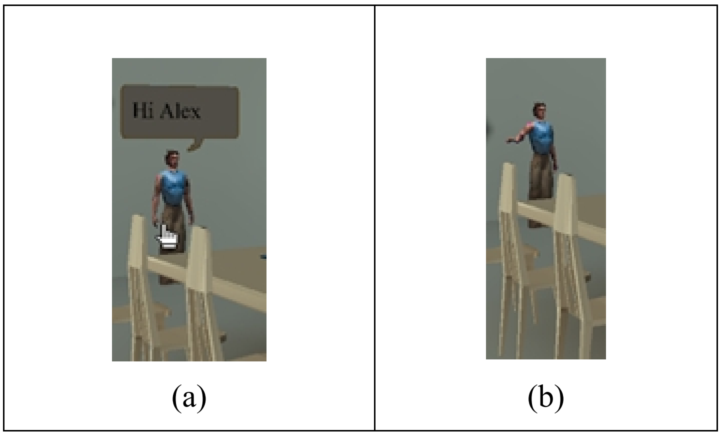
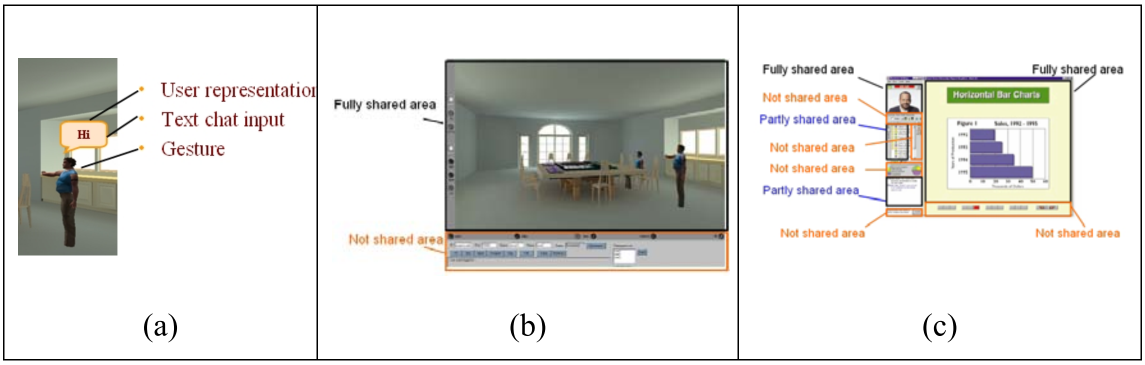


Figure 2: Examples of augmenting user's representation and awareness

Concerning awareness of objects and the action on them, there are many solutions. An example is depicted in Figure 7, where users can share and see the cards attached in the brainstorming board by their participants.

According to principle 4, the basic functionality of the interface should be accessible in a graphical user interface fashion in the context of a collaborative virtual environments. Furthermore, in order to reduce the amount of extraneous load of the users, EVE training area adopts the following approach:

- It integrates avatars with gestures. In such way the user can see at once who is participating and who is making what contribution. An example is depicted in Figure 3a.
- It separates the shared and not shared areas in order to avoid user's misconception as depicted in Figure 3b. A different design that could maximize the amount of extraneous load of the users is depicted in Figure 3c. In that case there are many areas that contain information fully, partly or not shared. Thus, the user could be overloaded in order to discover what the rest of participants are doing, who is participating, etc.



(a) (b) (c)

Figure 3: Design examples to reduce the extraneous load of the users

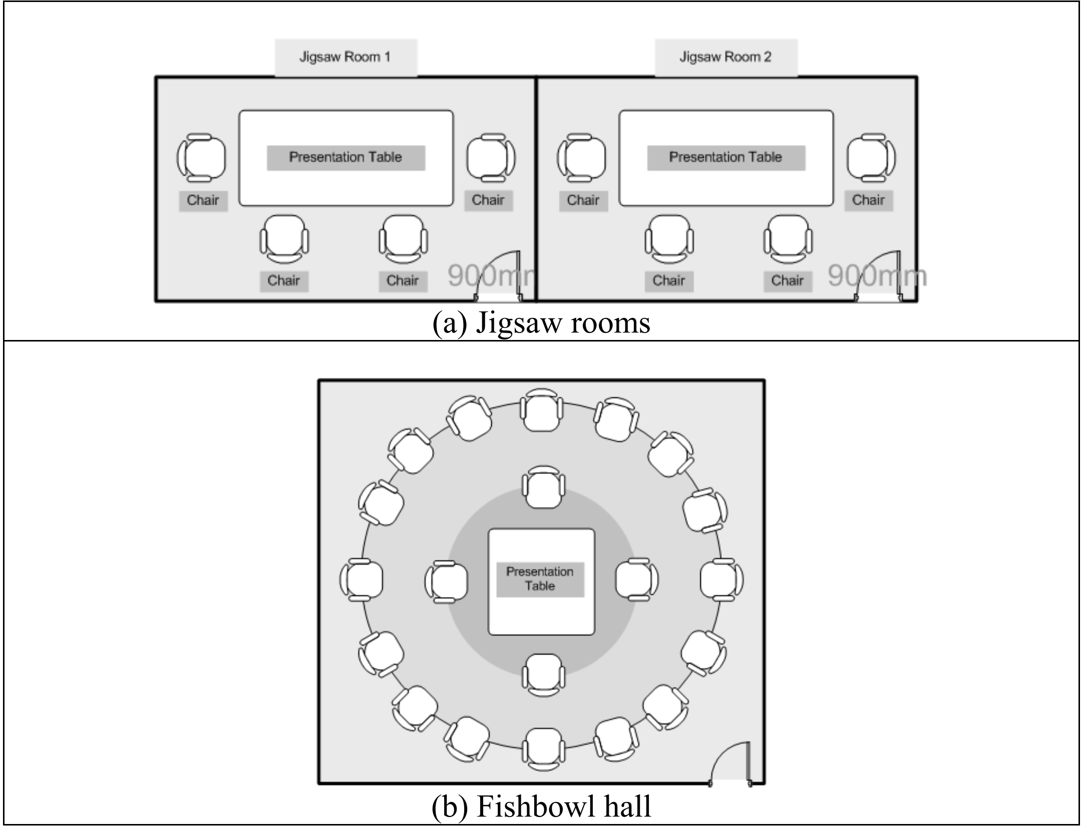
As previously described e-learning systems, supported by collaborative virtual environments, should be based on three main categories: Content, Learning Context and Communication Media (principle 5). The approach adopted in EVE training area with the concepts of (a) presentation table for sharing information; (b) avatars, audio conferencing and text chat for supporting communication; (c) 3D classroom design along with shared library for integrating learning content has been rated very positively as described in Bouras & Tsiatsos (2006). Thus such a design approach is proposed for supporting principle 5.

Limitations

EVE Training Area supports almost all the previous defined design principles. Thus, even if the use of virtual reality technology is not a required feature a priori, it seems that the use of collaborative 3D virtual environments and humanoid avatars along with supportive communication channels fit well as a solution for virtual collaboration spaces. Humanoid avatars are a unique solution that 3D-centered tools offer to group communication and learning. It is a fact that persons participating in the virtual learning experience with human like full-body avatars feel more comfortable than in chat or audio-communication (Bouras & Tsiatsos, 2006). The main benefit of the avatars is the psychological feeling of a sense of 'presence'. The sense of 'presence' results in a suspension of disbelief and an increase in motivation and productivity (Bouras & Tsiatsos, 2006). There are a number of important attributes to this experience. The ability to make basic gestures along with a voice or text message strengthens the understanding of the communication context (Redfern & Galway, 2002). Therefore, due to the fact that the user's awareness of the spatial proximity and orientation of others has a strong impact on the dynamics of group communication (Redfern & Galway, 2002), we could say that 3D multi-user virtual spaces have a good potential for supporting learning communities and e-collaboration. In such an environment users feel as though they are working together as a group and tend to forget they are working independently.

However, the previous versions of EVE should be improved in order to support the following functionality:

- *Multiple collaborative learning techniques (i.e. principle 1) and flexibility within a virtual space (i.e. principle 2)*: Even though, in the previous versions of EVE, it is feasible to implement and integrate various educational spaces in order to support different collaborative learning techniques, it is not possible to change on the fly the settings of the educational space. A comprehensive and thorough list of collaborative learning techniques is presented in Barkley et al. (2004). Examples are “fishbowl” (where the students form concentric circles with the smaller, inside group of students discussing and the larger, outside group listening and observing), “role play” (where, students assume a different identity and act out a scenario) and “jigsaw” (where, students develop knowledge about a given topic and then teach it to others). Depending on the set objective, the collaborative learning techniques can be used independently of, or in combination with each other. However, the spatial organization of the virtual environment could be very different for each technique. For example the jigsaw CL technique needs various rooms (which are furnished by chairs and a collaboration table) for supporting the discussion and collaboration among the members of jigsaw groups (Figure 4-a). On the contrary, fishbowl CL technique could be supported by a hall (Figure 4-b), which is furnished by chairs in two concentric cycles and a presentation board.



(a) Jigsaw rooms

(b) Fishbowl hall

Figure 4: Examples of virtual learning spaces for supporting collaborative learning techniques

Furthermore, in the previous versions of EVE platform is possible for the user (teacher) to use and choose various services/virtual tools for supporting the educational process. However, the tutor cannot reorganise the EVE training area in order to support better the learning needs as well as to avoid misunderstandings in the usage by the students. Due to the need for multiple functions within a collaborative online synchronous session, it should be possible to quickly reorganize the virtual place for a particular activity or scenario.

- *Spatial audio conferencing, which is targeted to support principle 3 (augmenting user's representation and awareness)*: The spatial audio conferencing support is an important feature of a networked virtual environment that aims to support either distance learning scenarios or e-collaboration (Begault, 1994). According to Barfield et al. (1997) the three primary benefits of auditory spatial information displays are identified as: (a) relieving processing demands on the visual modality; (b) enhancing spatial awareness; and (c) directing attention to important spatial events. Yamazaki & Herder (2000) refer that, by exploiting spatial audio conferencing, our eyes track a moving avatar on the screen and at the same time a spatialized sound source gives other information without disturbing the visual task. Furthermore, they claim that spatial audio conferencing can enhance the spatial awareness, because the spatial locations of all sound sources let us determine not only the location of other sound sources, but also our own location in space. They also claim that an acoustical event spatialized using a sound source can direct our attention. We tend to agree with the above claims and findings. Spatial audio is of equal importance with visual modality in order for an interactive 3D environment to be realistic. Plain audio conferencing support is an important feature. However, the best results are achieved by integrating spatial audio conferencing.

Functional Extensions

In the following paragraphs, we are presenting the necessary functional improvements in EVE platform for the support of:

- Multiple collaborative learning techniques, spatial collaboration on the implementation of CL techniques and flexibility within the virtual educational space
- Spatial audio conferencing

Multiple collaborative learning techniques and design to maximize the flexibility within a virtual space

EVE platform has been improved and offers a new module, giving the teacher the ability to design the EVE training area as s/he wants (Bouras et. al, 2007). From the users' side this module is a plug-in which is extended by a 2D tool (Figure 5) called "Collaborative Spatial Design Tool". This tool contains a number of panels to provide different functions. Besides the already existing panels from the previous versions of EVE platform (i.e. gesture, chat and lock panels), two new panels is introduced:

- The "*2D Top View Panel*" (Figure 5: 2D Top View Panel): This panel was embedded in the user interface as a tool for re-arranging worlds in collaborative spatial designs. It illustrates the floor plan of the world area and its objects. A user can move an object inside the limits of the world thus the limits of the panel and then s/he can watch the corresponding X3D object moving in the virtual X3D world. The introduction of this panel is of great importance. Not only it gives a better inspection of the object arrangement in the world, making it easier for the user to choose the modifications to be applied, but it also functions as lightweight object transporter. The events occurring on that panel are shared with the rest of the online users.

- The "*Options Panel*" (Figure 5: Options Panel): When dealing with collaborative spatial design options (such as object lists and classroom information) are a necessity. For that reason the option panel features options depending on the application. For example, this panel features options such as an object list for choosing virtual objects, a classroom object list, and number of copies of certain objects to be inserted, etc.

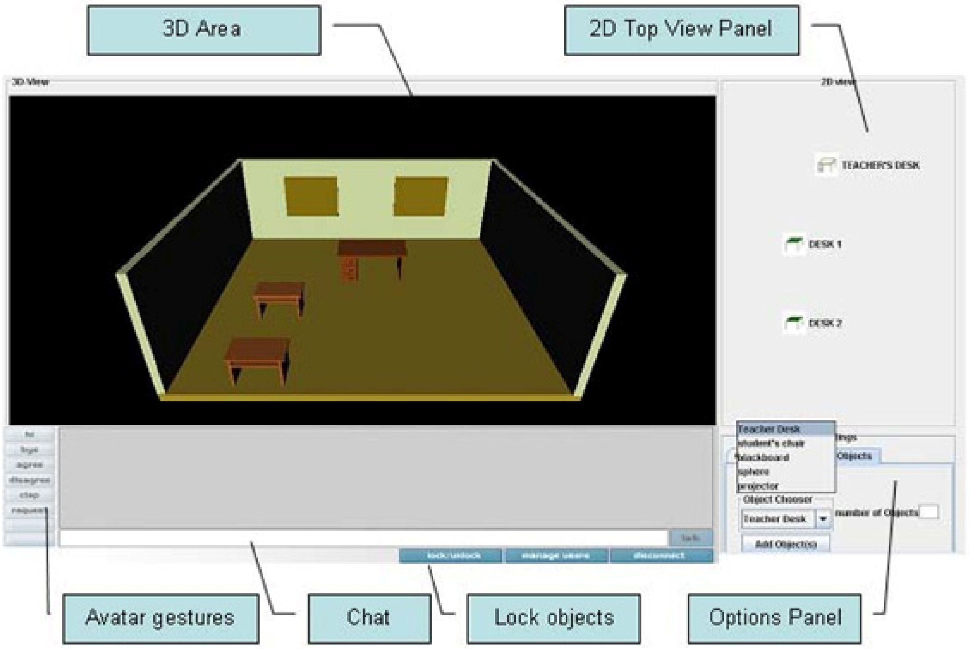


Figure 5: User Interface of EVE client extended with Collaborative Spatial Design Tool

Using the Collaborative Spatial Design Tool, depicted in Figure 5, the teacher can design the EVE training area as s/he wants by:

- using predefined classroom models and having the ability to reorganize the classroom
- creating and setting up of a virtual classroom using object library
- using spatial collaboration on the implementation of CL techniques

These usage scenarios are described in the following paragraphs.

Usage of predefined classroom models with classroom reorganization ability: This functionality offers quick classroom setup and the ability to move existing objects or to add new. The procedure that a teacher has to follow is to choose one of the predefined classrooms according to his/her criteria. Once the teacher selects a predefined classroom in which specific objects are placed has two options. The first one is to select new objects, which s/he wants to add, from an object list (Figure 5: Options Panel). The second one is to rearrange already added components. When a teacher loads a classroom a top view is created in a 2D panel next to the 3D world (Figure 5: 2D Top View Panel). Each 3D object has a 2D representation. The teacher can move an object in the 2D view. Accordingly the corresponding X3D object will be re-located in the virtual world. This scenario is preferred when the features and the customization needed for a classroom have to do mainly with objects' location and reorientation. In that case the avoidance of having to select an empty classroom and fill it with object saves much time.

***Creation and set up of a virtual classroom using object library*:** This functionality supports the teacher to implement multiple learning scenarios. More specifically s/he can change the organization of the virtual classroom and can use different shared objects that can facilitate each learning scenario. For example, s/he may want to select the size or shape of the virtual classroom, add specific objects etc. If that is the case EVE offers the ability to select from a variety of objects stored in a database library (Figure 5: Options Panel). Extended customization is offered by this model enhancing more precise customization. The teacher chooses an empty virtual classroom from a list of virtual classrooms, according to his/her needs. Then s/he adds the kind and number of objects s/he likes. Moving the 3D objects is supported as well, as described in the previous paragraph.

This functionality is giving the teachers the ability to select among a number of empty or already customized classrooms. A list of available objects can be added in the virtual classrooms by the teachers. Moreover, a teacher can move objects in the 2D floor plan. This plan contains a 2D representation of all the objects in the classroom. Dragging an object in the 2D view moves the corresponding object in the 3D world accordingly. For example, in order to apply the brainstorming/roundtable scenario the tutor can re-organise the classroom area by creating a table with a brainstorming board and seats for the learners around the table, as depicted in Figure 6.

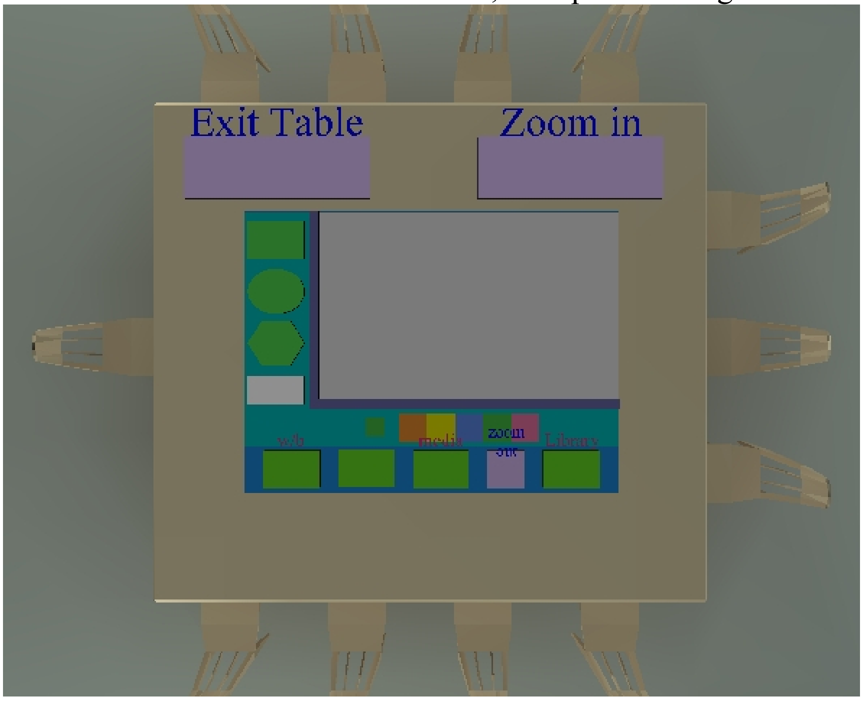
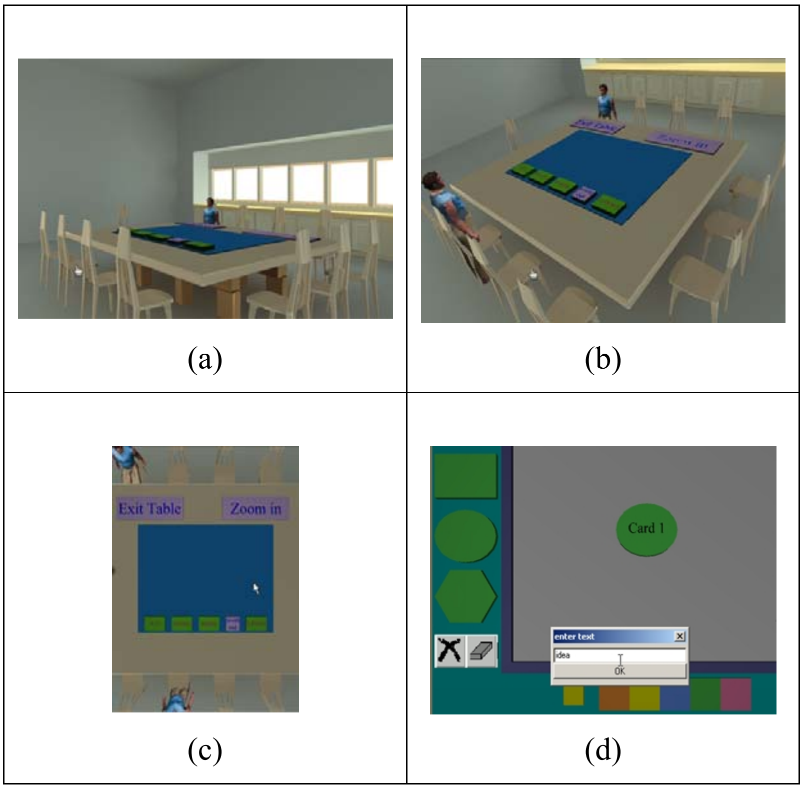


Figure 6: Organizing the EVE training area for brainstorming

EVE Training Area has been designed in such a way to maximise the flexibility within a virtual space (in order to satisfy principle 2). The tutor can reorganise the EVE training area in order to better support the learning needs as well as to avoid misunderstandings in the usage by the students. In that way, the tutor can either create or re-use virtual rooms for formal classes, group work, etc. For example in the organisation depicted in Figure 6 the only action for the user, in order to participate in the brainstorming session, is to move his/her mouse over a chair and to click on it (Figure 7a). By following these actions the viewpoint of the user is changed and s/he can see the presentation table and the other participants (Figure 7b and Figure 7c). After that the user can cooperate with the rest of participants in the brainstorming session by zooming in the brainstorming table (Figure 7d).



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

Figure 7: Brainstorming session

Spatial collaboration on the implementation of CL techniques: During the above scenarios an inexperienced teacher in the application of CL techniques could collaborate with an expert in order to rearrange the classroom. This collaboration can be supported by chat communication and 3D objects sharing. Furthermore, the expert can take the control to organize the classrooms adding and rearranging 3D objects. Alternatively, two or more teachers can collaborate concerning the creation and organization of a virtual classroom.

Spatial audio conferencing

The integration of spatial audio conferencing in CVE platforms could facilitate the users' communication in terms of immediateness, while at the same time the voice contributes to a more realistic communication and interaction among the users. Spatial audio conferencing contributes to a best perception of the environmental entities, especially when the user has no eye contact with them. The user, by hearing spatial audio, obtains information about the 3D location, and the direction of the entity which emits the sound moves. Moreover, depending on the intensity and the tone of the sound, a user can be aware, to some extent, of the intensions of the entity towards the user as well as the psychological situation of the entity. These psychological effects that arise from the 3D spatial sound along with the perception of space, lead to a very realistic interaction in two fashions: among the users, and between the user and the virtual space.

Technological and implementation issues

The advancements presented in this paper were implemented in EVE Networked Virtual Environments platform. Thus, it is essential to present the main characteristics of this platform. EVE is based on open technologies (i.e. Java and X3D). It features a client-(multi) server architecture (Figure 8) with a modular structure that allows new functionalities to be added with minimal effort. The previous version of EVE platform provided a full set of functionalities for e-learning procedures, such as avatar representation, avatar gestures, content sharing, brainstorming, chatting etc.

However, the above described functionality for maximizing the flexibility within a virtual space, implies the implementation and integration of a collaborative spatial design application in EVE platform. Such an application requires:

- a flexible way for dynamic 3D world manipulation such as the ability to dynamically load virtual environments and shared objects.
- a generic event's management mechanism for effective non-X3D event handling.

From the technical point of view X3D (Web3D, 2009) is the current open standard for lightweight 3D content description and representation, to build virtual environments. It can be used from desktop to web applications and is the ideal solution for open platforms. However, X3D standard does not support 3D event sharing, which is necessary for the multi-user nature of Collaborative Virtual Environments. The previous version of EVE integrated an X3D event-handling mechanism responsible for serving events related to the virtual world. However, a more robust and complete solution has been implemented in the extended version of EVE platform in order to accomplish the demand for dynamic X3D node loading.

Concerning the integration of the Collaborative Spatial Design Tool, there is a need to handle non-X3D events in order to support:

- the retrieval of new 3D objects or whole virtual worlds from a database, (such as database queries to retrieve objects and 3D environments from a database),
- the manipulation of 3D objects from an external and intuitive 2D interface, which implies the support of java swing events.

These events are called "2D applications events". An additional server called "2D application server" has been developed and integrated in EVE platform for servicing "2D applications events".

Concerning the integration of spatial audio conferencing, EVE platform featured H.323 audio conferencing since it was introduced. However in order to support the spatial audio conferencing a new Audio Conferencing Server is needed. This server features a new algorithmic approach and utilizes the latest session protocols and codec technologies.

The main issues concerning the integration of spatial audio conferencing are the following:

- the selection of a networked audio protocol along with algorithms and codecs that will be used to establish sessions and reproduce sound among users
- the algorithms that will create the illusion of 3D spatial sound

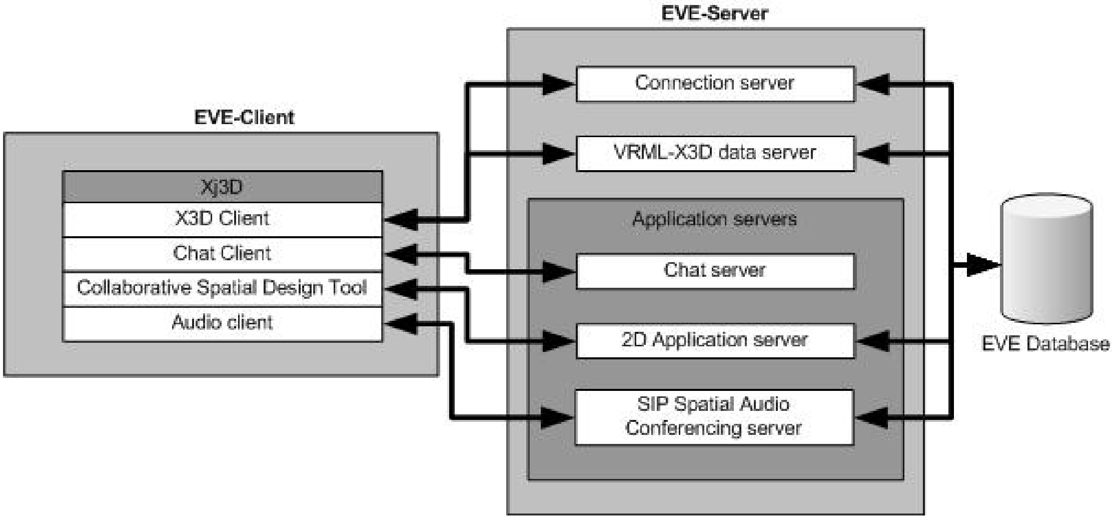


Figure 8: EVE architecture

After the integration of the new features, the client-side of EVE platform is a Java applet that incorporates an X3D browser (based on Xj3D API), and audio and a chat client.

The server-side architecture of the EVE platform consists of five servers as shown in Figure 8. The “Connection Server” coordinates the operation of the other three. The “VRML-X3D Server” is responsible for sending the 3D content to the clients as well as for managing the virtual worlds and the events that occur in them. The “Chat Server” supports the text chat communication among the participants of the virtual environments. The “SIP Spatial Audio Conferencing Server” is used to manage audio streams from the clients and to support spatial audio conferencing. Finally the “2D Application Server” handles the generic (i.e. non-X3D) events.

Both “SIP Spatial Audio Conferencing Server” and “2D Application Server” are the new servers that integrated in EVE platform. More details about this server and the related implementation issues will be described in detail later in this paper.

The rest of the paper is focused on presenting the related work and the work done concerning the technological advancements of EVE platform.

Related Work on Technological Issues

This section presents the related work in point of the advancements of EVE platform. Therefore, it is organized in the following parts describing work done on: (a) X3D enabled NVE platforms; (b) protocols for supporting 3D event sharing in NVEs; (c) spatial audio conferencing; and (d) collaborative design applications.

X3D enabled NVE platforms

This paragraph presents an overview of the state of the art on X3D enabled networked virtual environment platforms. 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 (http://www.blaxxun.com), Bitmanagement’s BS Collaborate (BS Collaborate, 2007), Active Worlds (http://www.activeworlds.com), Octaga (http://www.octaga.com), ParallelGraphics (http://www.parallelgraphics.com), Croquet (http://www.croquetconsortium.org), I-maginer, (http://www.i-maginer.fr/), Second Life (http://secondlife.com) and Workspace 3D (http://www.tixeo.com). The most significant research platforms are the following: DIVE: Distributed Interactive Virtual Environments (Carlsson & Hagsand, 1993; http://www.sics.se/dive), SPLINE: Scalable Platform for Large Interactive Environments (http://www.merl.com/projects/spline), VLNET: Virtual Life Network (Pandzic et al. 1996; Pandzic et al. 1998), SmallTool (Broll, 1998) and EVE (Bouras et al. 2005; Bouras et al. 2006; http://ouranos.ceid.upatras.gr/vr).

However, some of the above platforms are not supporting X3D standard at all and some of them are supporting X3D standard partially (Bouras et al. 2005). The most promising X3D enabled CVE platforms today are Bitmanagement, Octaga and EVE solutions. Almost all these platforms partially support X3D standard, and offer good rendering functionality. However, the first two (i.e. Bitmanagement, and Octaga) 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. Therefore, 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. Thus, a promising technical solution could be EVE platform. For that reason we have decided to work on the extension of EVE platform in order to accomplish the demand for dynamic X3D node loading.

Protocols for supporting 3D event sharing in NVEs

This paragraph is presenting briefly the main protocols for handling 3D event sharing in NVEs, in order to adopt similar mechanisms (if any) for dynamic X3D node loading. These protocols are VRTP (Virtual Reality Transfer Protocol) (Brutzman et al. 1997), DIS (Distributed Interactive Simulation) protocol (Canterbury, 1995) and SWAMP (Simple Wide Area Multi-user Protocol) (Weber & Parisi, 2007).

VRTP is an application level protocol for Internet based NVE's in a standardized way. It offers four basic functionalities for NVEs communication capabilities: (a) entity state processing, (b) heavyweight objects, (c) network pointers and (d) real-time streams. VRTP framework consists of a protocol collection and an application level protocol that provides the necessary connectivity between the client and the virtual environment. The main problem of VRTP is that its design is focused on the support of the communication needs of VRML-based NVEs; It wasn't designed in order to meet specific networking demands.

The DIS protocol has been designed to support large scale Virtual Environments and is based on the SIMNET standard (Miller & Thorpe, 1995). It consists of a set of protocols. More specifically DIS defines a set of Protocol Data Units (PDUs), which are transmitted to all the participants of the NVE in order to update the current state of each object. DIS is very efficient in supporting many concurrent users. However, it is difficult to be integrated in EVE platform because of its limited scalability and the rigidity caused by embedding its application in its architecture (Wray & Hawkes, 1998).

Finally, SWAMP is a new multi-user protocol which was designed in order to support X3D client-server communication. The philosophy behind its design is the ability to be used in wide area, thus heterogeneous, networks. A great amount of concern has been given to the message exchange speed and safety. SWAMP uses an entity model that is based on the X3D rationale (i.e. nodes and fields). Abstractions of an entity or a total of entities are also supported. The approach of SWAMP for the client-server communication, regards the use of TCP/IP for the establishment of the initial connection as well as for the exchange of low frequency messages and utilizes UDP for exchanging continuous, highly frequent messages. UDP usage involves the use of packages with small overhead, which results in faster transportation. SWAMP integration in the X3D scene graph involves the usage of the X3D Network Component. This features a node named EventStreamSensor that establishes network connections. However, SWAMP is still under development and thus it is not ready for public use.

Our work differs from SWAMP in the type of the events that are sent to the server. Instead of using the EventStreamSensor node to transmit events, each event that occurs in the virtual world is captured by a custom event sharing mechanism and is transmitted to a dedicated server for processing. The mechanism of our solution is more generic, since it allows the sharing of events that occur outside the 3D scene as well. By that way our mechanism can support the dispatch of an event of an external application to the 3D scene. In addition our mechanism is mature and fully functional, offering a variety of events that is a superset of the X3D standard set of events.

Spatial audio conferencing

A fair amount of work has been done on spatial 3D sound. The majority of today's 3D games, single or multi-user, features spatial 3D sound. However the sound that is used in this type of applications is pre-recorded. When it comes to CVE's where live streaming sound needs to be converted to spatial 3D sound, little work has been done. Good examples have been presented by Liesenborgs (2000) and Macedonia et al. (1995).

The work done by Macedonia et al. (1995) is based on multicast networks. It features a networked virtual environment that offers low cost 3D sound. We want to avoid this solution due to the fact that multicasting is not available in every network.

Liesenborgs (2000) describes a Voice over IP framework for networked virtual environments. The capture and reconstruction of voice is accomplished by the clients' operating system. The session initiation and the transmission of the voice are carried out by Real-time Transmission Protocol (RTP) and Session Initiation Protocol (SIP) (RFC3261, 2002) libraries written in C++. The spatialization of the audio is performed by an algorithm that mimics the way human ear perceives sound, that is using the Interaural Time Difference (ITD) and the Interaural Intensity Difference (IID) effects. Our work differs from the work done by Liesenborgs (2000) in terms of portability, since our solution is Java based thus platform independent. The second and major difference is that our solution relies on an X3D standard node to perform the audio spatialization in X3D based virtual environments.

Spatial collaboration applications

There is some research work done on spatial collaboration using CVE. This work, as described in the following paragraphs, is focused on: (a) the usefulness of multiple representations and the need for additional features to support collaboration across representations; (b) the viewpoints handling; (c) findings concerning intuitiveness and real-time interaction issues in virtual environments for supporting spatial collaboration.

The main example is the work done by Schafer & Bowman (2005), which investigates how to support distributed spatial collaboration activities and presents a novel prototype that integrates both two-dimensional and three-dimensional representations. Our work differs from this project in terms of technological solution as well as from the communication channels included. Our scope is to rely on the use of the combined representations and the findings of Schafer & Bowman (2005), which highlight the usefulness of multiple representations and the need for additional features to support collaboration across representations.

Another example is CALVIN prototype (Leigh & Johnson, 1996; Leigh et al. 1996). This system explores the usage of different virtual reality hardware configurations, such as CAVE and Immersadesk technology, for collaboration. Although our scope is to design and develop a system for desktop CVEs, using only keyboard and mouse as input devices, the findings of this work are useful concerning the viewpoints handling.

Another interesting work concerns VSculpt (Li & Lau, 2003), which is a collaborative virtual sculpting system that enables geographically separated designers to participate in the design process of engineering tools and sculptures. Although this work aims at supporting collaborative virtual sculpting, its findings concerning intuitiveness and real-time interaction are very useful for the extension of our platform.

Furthermore, Li et al (2006) implemented a 3D collaborative system for double-suction centrifugal pump, based on X3D and Java Applet technology. The system allows the assembly of a centrigugal pump by one or more collaborators via a web interface that is embedded in an X3D browser. Our work is different in the way that the 3D objects are externally modified, and that user can move objects inside the 3D scene. Our work is based on the use of a two dimensional ground plan of the scene that displays labels of all the available 3D objects. These labels can be dragged within the 2D ground plan resulting in the movement of the corresponding 3D objects in the 3D scene.

X3D Support and Event Sharing

This section presents two important features of the EVE platform:

- The support of X3D standard, which is the current standard in the area web based virtual reality applications.
- The extension of X3D by a custom event sharing mechanism, which manages and shares events over the network.

Generic X3D Features

Originally, VRML (Virtual Reality Modeling Language) was used in EVE for 3D content creation and visualization. VRML evolved to, the ISO certified, X3D open standard. X3D supports XML encoding as well as the syntax of the VRML language. Many advantages derive from the use of XML, such as interoperability with other networking applications and familiar syntax to web applications developers. Considering that EVE is a web based platform, the use of an XML based open standard is very important. Moreover, X3D allows lightweight core 3D runtime delivery engine. This is crucial since EVE's client runs within a browser where minimum consumption of resources is required. Furthermore, there is no tradeoff between lightness and quality since X3D graphics are of high quality, while performing in real-time.

In order to dynamically modify 3D content, which is a key feature for the interactive virtual environments, the Scene Authoring Interface (SAI) was created. SAI is the appropriate API (Application Programming Interface) for user-3D scene interaction. Using SAI programmers can add, remove and modify nodes and their fields from both inside and outside the 3D scene.

Although X3D offers many advantages for visualizing and interacting with web based virtual environments, there is no mature or cost-effective solution for multi-user event sharing over the network, as presented in a previous section. Therefore, the only solution for supporting multi-user virtual environments is the extension of X3D by implementing an event sharing mechanism. This mechanism is described in the next paragraph.

X3D event sharing mechanism

An NVE is based on a mechanism for sharing events that occur in the virtual scenes. This is important in order to maintain 3D content consistency and to allow interaction among users. As said before, X3D currently does not provide a mechanism and/or protocol to share 3D scene events. The event sharing mechanism of the previous version of EVE platform is now improved. In order to accomplish the demand for event sharing support of dynamic created 3D objects (i.e. dynamic X3D node sharing and loading). The previous version of EVE integrated an X3D event-handling mechanism responsible for serving events related to the virtual world. The previous mechanism overrides SAI and EAI (External Authoring Interface) in a way that events are sent to all users connected to the platform. In order to dynamically create nodes a specific event is sent to the VRML-X3D server, containing the node to be added and the parent (default is "root node") to make this node its child. This event is then broadcasted to online users and is added to an X3D representation of the world it belongs. This representation is kept in the server and it is broadcasted to new users that entered later in the virtual world. It should be pointed out that already online and connected users to the platform receive only the newly added nodes. By that way, networking load is significantly reduced. Once online clients receive a shared node they locally add it to their VRML-X3D Scene.

In general, the design of event sharing mechanism is trying to fulfill the following three requirements:

- event sharing support of many data types and events
- easy transformation of a non multi-user X3D virtual world to a multi-user one, based on little code changes to the initial X3D file
- selection of efficient and suitable network protocols for good network bandwidth management.

In order to satisfy the above requirements, the event sharing mechanism features an internal Java representation of X3D nodes and fields. Events that are needed to be shared are marked with a "shared" tag, in the corresponding field routes. This method causes a minimum change to the initial X3D file. When a new shared event occurs in a client's virtual world the following steps are followed:

- The event is being transmitted to the VRML-X3D server without affecting the local copy of the virtual world in the client.
- The event is converted to an instance of class responsible for describing event, with all the necessary parameters stored as well.
- The event is processed by the server and is sent back to the clients that the event concerns.
- The event is received by the client, it is transformed via SAI to reflect the change inside the virtual scene.

The network protocol used for the event transmission is, generally, TCP in order to ensure reliability. However, the events occurring from avatars' position or orientation changes are transmitted via UDP. This ensures low network load, while at the same time it does not have impacts on the quality of the user's experience, since packages of such events are very frequently transmitted.

Implementation issues

The implementation of the above described mechanism involves both 3D content handling and parsing. As EVE is mainly implemented in Java, the Xj3D toolkit is used for the X3D and SAI implementation. We have extended Xj3D's Java library by creating a custom library (called "vrmlx") in order to provide support for shared events. This library contains packages that describe the fields and nodes of the X3D standard (Web 3d, 2007). Every vrmlx node extends a basic node named BaseNode. Similarly, each field extends a basic field named "Field". Each time occuring an event in the virtual world, the event's attributes related to X3D (such as a node or a field) are transformed to a vrmlx library representation. Afterwards, the clients receive the event, convert its vrmlx representation to an Xj3D representation and the event is applied to the 3D scene.

In order to support X3D virtual world loading and dynamic X3D objects' events sharing an X3D parser has been implemented. This parser parses the x3d files and loads the 3D content. The parser has been integrated in EVE platform as a package (called "vrmlx.parser") in the vrmlx library.

EVE'S SIP Spatial Audio Conferencing Server

The previous versions of EVE platform featured H.323-based audio conferencing. In order to increase performance and stability as well as to emphasize the distributed nature of the platform a new Audio Conferencing Server was introduced. This server features a new algorithmic approach and utilizes the latest session protocols and codec technologies.

The main issues concerning the integration of spatial audio conferencing are the following:

- the selection of a networked audio protocol along with algorithms and codecs that will be used to establish sessions and reproduce sound among users
- design and implementation of the algorithms that will create the illusion of 3D spatial sound

These issues are described in the following paragraphs.

EVE audio spatialization process and conferencing architecture

The steps of the audio spatialization process adopted in EVE platform are the following:

- The establishment of the client-server connection
- The capture of the audio stream by the client
- The transmission of the audio stream to the server
- The spatialization process of the audio stream in the virtual world
- The reception of the audio stream by the rest of clients

The technologies used in order to support the above process are the following:

- Session Initiation Protocol (SIP) for the session establishment. This lightweight, transport independent protocol has proven to be very reliable and robust thus making it the most popular session protocol nowadays
- Real Time Protocol (RTP) for audio data transmission (RFC3550, 2003).
- The Java Media Framework API, which provides convenient classes and methods for media manipulation, is used for supporting the audio capture.

Concerning the process for audio spatialization the proposed and implemented solution takes into account issues such as bandwidth consumption, processing costs, complexity, as well as design and implementation issues related to EVE platform.

Audio spatialization is performed by an X3D Sound node (Figure 9). Firstly, the client's side architecture is examined. The establishment of the client-server connection is accomplished via the SIP protocol. The client's applet makes a call to the SIP Spatial Audio Conferencing Server and a server port is reserved for the connection with the client. After the session initiation, an RTP stream, using the JMF API, is established with the server, in order to transmit the audio data stream. At the same time the client's capture device captures audio data, utilizing JMF API's classes, and transmits them through the RTP stream. The X3D browser of the applet receives the audio data encapsulated, by the SIP Spatial Audio Conferencing Server, in X3D AudioClip nodes. The playback of the audio is performed by X3D Sound nodes that use the AudioClip nodes as their sources.

The X3D Sound node features built-in spatialization and attenuation audio algorithms. Regarding the server side, the following procedure takes place: The server is waiting for new SIP calls on a dedicated port. After an incoming call from a client is accepted, a new port is assigned for the communication between a server thread, dedicated in servicing the specific client, and the server. This thread establishes an RTP stream with the client for receiving audio data. Concurrently, the SIP Spatial Audio Conferencing Server acquires information of the user's avatar location and orientation in the virtual world through the VRML-X3D Server. This information will be used to reproduce the sound like it is being emitted from the avatar's mouth. For each user an X3D AudioClip node along with an X3D Sound node are instantiated via the Xj3D API and they are added in the graph scene of the virtual world. A file is created to which the audio data are continuously appended to. The AudioClip node's URL field is given that file as a value, while the Sound node's fields direction, location and source fields are given the values of the avatar's mouth direction, avatar's mouth location and the AudioClip, respectively. The key point in this solution is that the audio spatialization is based on the X3D Sound node. This node can produce spatialized audio by setting appropriate values to the specialized fields.

When the two new X3D nodes are added to the scene, the VRML-X3D server sends them to the client and the client's X3D browser starts immediate playback of the Sound node.

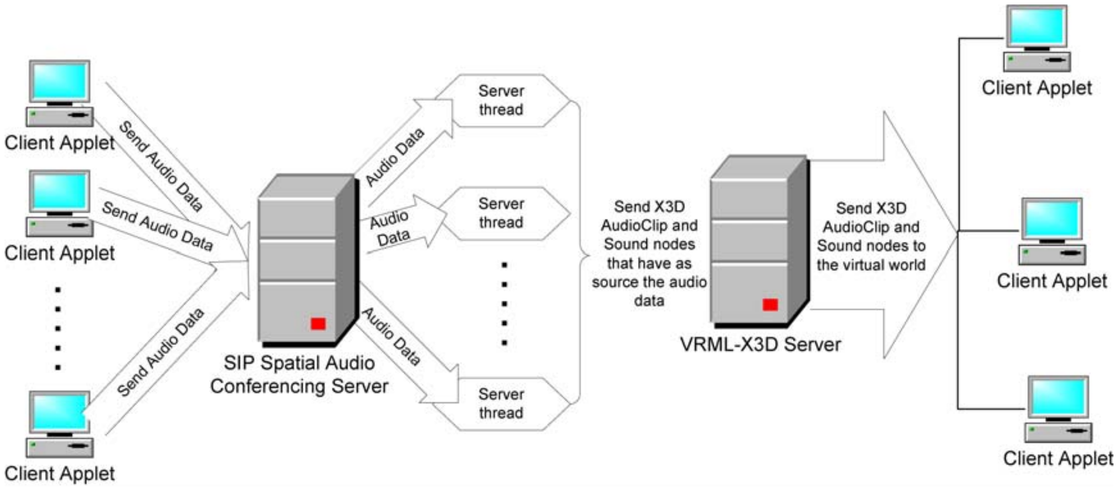


Figure 9: Audio conferencing architecture

Implementation Issues

In this section the implementation issues (SIP integration, audio streaming, capture and spatialization) are discussed along with the main technologies in which the SIP Spatial Audio Conferencing Server is based on.

- SIP integration: A java implementation of the protocol was adopted due to the fact that the EVE platform is Java – Xj3D based. More specifically, JAIN – SIP (JAIN – SIP, 2007) was chosen as an API. This solution helps us to maintain EVE’s openness and cross platform characteristics.
- RTP streaming and audio capture: The RTP streaming and the audio captured management tasks, is performed by custom code that utilizes the *Java Media Framework API* – JMF. The capture format that is used in this implementation is in linear encoding, has 8000 Hz sample rate, 8 bits of sample size and is monophonic. The streaming format was of the same characteristics. We used these relatively low quality settings in order to save bandwidth
- Audio spatialization: The spatialization of audio is performed by the X3D Sound Node. The Sound node specifies fields that affect the spatialization of the sound. The sound is located at a point in the local coordinate system and it is emitted in an elliptical pattern. The location is specified by the *location* field, while the direction vector of the ellipsoids is specified by the *direction* field. There are fields that specify the maximum and minimum values to where the sound is audible, that is the maximum and minimum lengths of the two ellipsoids along the direction vector. A very crucial field is the *spatialize* field. If set to *TRUE* the sound is perceived as being directionally located relative to the listener. If the listener is located between the transformed inner and outer ellipsoids, the listener's direction and the relative location of the Sound node is taken into account during playback. In our implementation this field is set to *TRUE*, resulting in a very realistic spatialized audio playback. The sound source specified by the field source is an AudioClip node. The AudioClip node specifies an URL field (that in our implementation is the URL file of the buffer), which is used as source. In order to change between the two sets of nodes, we used an ECMA script which sets the *startime* field of one AudioClip that waits to start, equal to the *stoptime* field of the currently playing AudioClip.

Generic Event’s Management: Eve’s 2D Application Server

As described before, there is a need to handle non-X3D events in order to support:

- the retrieval of new 3D objects or whole virtual worlds from a database, (such as database queries to retrieve objects and 3D environments from a database),
- the manipulation of 3D objects from an external and intuitive 2D interface, which implies the support of java swing events.

These events are called “2D applications events”. Five types of “2D application events” are currently supported:

- SQL Database query (which is a string representing an SQL query)
- JDBC ResultSet (a JDBC ResultSet class)
- Java Swing Component (such as labels, shapes, etc.)
- Java Swing Events (such as altering the location of a Swing Component)
- Ping: Used to verify the connection between the server and the clients

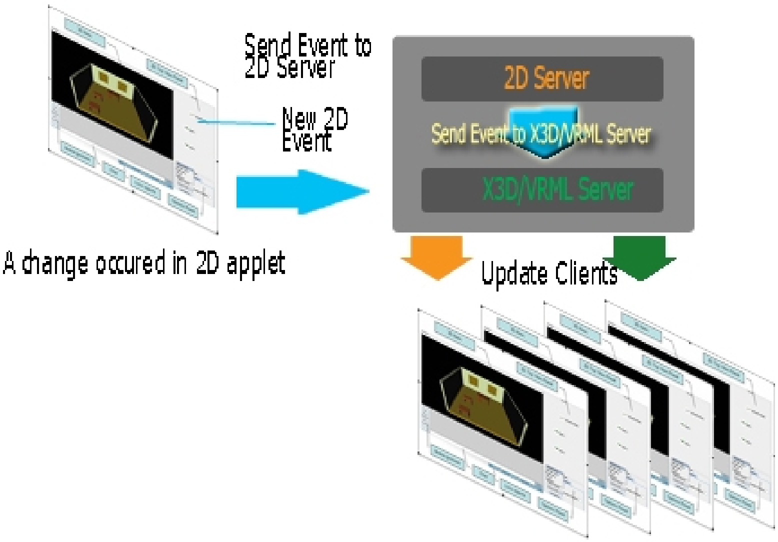


Figure 10: 2D Events Architecture

EVE, as already mentioned, consists of several servers. However, none of them is able to manage 2D events. For that reason an additional server called "2D application server" has been integrated in EVE platform.

The client-server communication operation is described as follows (Figure 10): Firstly a client establishes a connection to the server. This class deals with connection related issues such as sockets server ports etc. Once a connection has been established, two threads (one responsible for sending and one for receiving "2D application event" instances) are created for each client. On the client side, a thread responsible for "2D application event" handling and server communication is created. The receiving thread examines if the event is to be executed in the server (e.g. Database query). In that case, it executes the event and, if necessary, it creates another event (e.g. ResultSet). Otherwise, it enqueues the event in the ClientConnection FIFO queue. After that the sending thread takes the first pending event and sends it to all clients.

Future Work

Among our next steps, it is to extensive test the newly added features among a various usage scenarios by large groups of users. Also, several technical aspects of the features need to be thoroughly examined, such as scalability, packet jitter and server load.

In addition, we plan to extend EVE platform in order to support mixed reality applications. Mixed reality applications usually involve the usage of specialized and expensive equipment in order to mix real world with virtual world (i.e. computer generated data). This can be done by utilizing the generic event's management mechanism to handle non-3D data such as video projections.

Concerning spatial collaboration applications, our next step has mainly to do with extended world setup abilities. Particularly, a user will have the abilities to add his/her custom X3D objects, change a classroom's dimensions, and visualize possible collisions. Collisions may occur due to the following reasons: (a) specific spatial setup models; (b) accessibility to emergency exits in case of an emergency situation; (c) routes a teacher follows during class time; and (d) students co-existence problems.

Also, we intend to provide full H-Anim (http://www.h-anim.org/) support in EVE avatars. The H-Anim protocol is a standard way of modeling and animating humanoids, designed to facilitate the sharing of animations between different humanoid models. This will allow the creation of portable avatars that can be transferred from one NVE platform to another. By supporting H-anim standard we will facilitate the users to upload and use their custom H-Anim compliant avatar.

Finally, we plan to evaluate this platform exploiting the framework suggested by Tsiatsos, Konstantinidis and Pomportsis (2010). This framework consists of two consecutive cycles. Each cycle is made up of individual phases consisting of three steps: main phase step, data acquisition step and data analysis step. There are three types of phases: pre-analysis, usability and learning. This evaluation will include the Jigsaw CL technique, the Fishbowl CL technique and the participation of postgraduate students.

Conclusions

This paper presented advanced functionality that has been integrated on "EVE Training Area tool" in order to support: (a) multiple collaborative learning techniques (b) Spatial audio conferencing, which is targeted to support principle 3.

Furthermore, the paper presented technological and implementation issues concerning the evolution of "EVE platform" in order to support this functionality.

EVE platform is based on open technologies (i.e. Java and X3D). It features a client – multi-server architecture with a modular structure that allows new functionalities to be added with minimal effort. The previous version of EVE platform provided a full set of functionalities for e-learning procedures, such as avatar representation, avatar gestures, content sharing, brainstorming, chatting etc. However, the above described functionality for maximizing the flexibility within a virtual space, implied the implementation and integration of a collaborative spatial design application in EVE platform. Such an application required: (a) a flexible way for dynamic 3D world manipulation such as the ability to dynamically load virtual environments and shared objects; and (b) a generic event's management mechanism for effective non-X3D event handling. For that reason the X3D standard has been extended by a custom event sharing mechanism over the network, which manages and shares events that occur in the virtual worlds. This mechanism has been integrated in EVE platform.

Another important extension of EVE platform is the design and implementation of a spatial audio conferencing server to support spatial audio conferencing functionality. By that feature EVE can support the three primary benefits of auditory spatial information displays that are identified as: (a) relieving processing demands on the visual modality, (b) enhancing spatial awareness, and (c) directing attention to important spatial events. Concerning the integration of spatial audio conferencing we have faced the following issues: (a) the selection of a networked audio protocol along with algorithms and codecs that will be used to establish sessions and reproduce sound among users; and (b) the design and implementation of the algorithms that will create the illusion of 3D spatial sound. As far as the protocol is concerned, the Session Initiation Protocol (SIP) was used. The transmission of the audio data utilizes the Real-time Transport Protocol (RTP)

The last technological improvement in EVE platform was the integration of a module for handling non-X3D events in order to support (a) the retrieval of new 3D objects or whole virtual worlds from a database, (such as database queries to retrieve objects and 3D environments from a database); and (b) the manipulation of 3D objects from an external and intuitive 2D interface, which implies the support of java swing events.

An additional server called “2D application server” has been developed and integrated in EVE platform for servicing this type of events.

To conclude with, the new version of EVE platform, by integrating the above features can support in a more effective way the EVE training area by offering to its users new tools and flexibility concerning the application of collaborative learning techniques. Now the teacher can design the EVE training area as s/he wants by (a) using predefined classroom models and having the ability to reorganize the classroom; (b) creating and setting up a virtual classroom using object library; and (c) using spatial collaboration on the implementation of CL techniques.

References

Barfield, W., Cohen, M. & Rosenberg, C. (1997). Visual, Auditory, and Combined Visual-Auditory Displays for Enhanced Situational Awareness. *The International Journal of Aviation Psychology*, 7(2):123–138.

Barkley, E., Cross, P. & Major, C. (2004). *Collaborative Learning Techniques: A Handbook for College Faculty*, Jossey-Bass, ISBN-10: 0787955183

Begault, D. R. (1994). *3-D Sound for Virtual Reality and Multimedia.* Academic Press Professional, Inc.

Bouras, C., Giannaka, E. & Tsiatsos, Th. (2008). Exploiting Virtual Environments to Support Collaborative E-Learning Communities, *International Journal of Web-Based Learning and Teaching Technologies,* IDEA GROUP PUBLISHING, Vol. 3, Issue 2, C. Bouras, E. Giannaka, T. Tsiatsos, 2008, pp. 1 - 22

Bouras, C., Giannaka, E., Panagopoulos, A., Tsiatsos, T. (2006). A Platform for Virtual Collaboration Spaces and Educational Communities: The case of EVE. *Multimedia Systems Journal, Special Issue on Multimedia System Technologies for Educational Tools*, Springer Verlang, Vol. 11, No. 3, pp. 290 – 303.

Bouras, C., Panagopoulos, A. & Tsiatsos, T. (2005). Advances in X3D multi - user virtual environments. *IEEE International Symposium on Multimedia (ISM 2005)*, Irvine, California, USA, 12 - 14 December, 2005, pp. 136 – 142.

Bouras, C., Psaroudis, C., Psaltoulis, C. & Tsiatsos, T. (2001). A Platform for Sharing Educational Virtual Environments. *9th International Conference on Software, Telecommunications and Computer Networks* (SoftCOM 2001) Split, Dubrovnik (Croatia) Ancona, Bari (Italy), Vol II, 9 - 12 October 2001, pp. 659 – 666.

Bouras, C. & Tsiatsos, T. (2004). Distributed Virtual Reality: Building a Multi - User Layer in EVE Platform. *Journal of Network and Computer Applications*, Academic Press, Vol. 27, No. 2, April 2004, pp. 91 – 111.

Bouras, C., & Tsiatsos, T. (2002). Extending the Limits of CVEs to Support Collaborative e-Learning Scenarios. *2nd IEEE International Conference on Advanced Learning Technologies*, Kazan, Russia, 9 - 12 September 2002, pp. 420 – 424.

Bouras, C. & Tsiatsos, T. (2006). Educational Virtual Environments: Design Rationale and Architecture. *Multimedia Tools and Applications*, Kluwer Academic Publishers, June 2006, Vol. 29, No. 2, pp. 153 – 173.

Bouras, C., Tegos, C., Triglianos, V. & Tsiatsos, T. (2007). X3D multi-user virtual environment platform for collaborative spatial design. The 9th International Workshop on Multimedia Network Systems and Applications (MNSA-2007), Toronto, Canada, 25 - 29 June 2007.

Broll, W. (1998). 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, pp. 118-128.

BS Collaborate (2007). BS Collaborate documentation. Bitmanagement Software GmbH. Retrieved September 10, 2009, from http://www.bitmanagement.com/download/BS_Collaborate/BS_Collaborate_documentation.pdf

Brutzman, D., Zyda, M., Watsen, K. & Macedonia, M. (1997). Virtual reality transfer protocol design rationale. *The IEEE Sixth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'97), Distributed System Aspects of Sharing a Virtual Reality workshop*, 179-186, IEEE Computer Society, Cambridge, MA, June 1997.

Canterbury, M. (1995). *An Automated Approach to Distributed Interactive Simulation (DIS) Protocol Entity Development*. Master's Thesis, Naval Postgraduate School, Monterey, California, September 1995.

Capin, T., Pandzic, I., Magnenat-Thalmann, N. & Thalmann, D. (1999). Avatars in Networked Virtual Environments. John Wiley & Sons Ltd, ISBN 0-471-98863-4.

Carlsson, C. & Hagsand, O. (1993). DIVE: a Multi User Virtual Reality System. *IEEE 1993 Virtual Reality Annual International Symposium*, VRAIS '93, Piscataway, NJ: IEEE Service Center, pp. 394.

JAIN – SIP (2007). Java API for Signaling. Retrieved September 10, 2009, from https://jain-sip.dev.java.net

Koubek, A. & Müller, K. (2002). Collaborative Virtual Environments for Learning. *ACM SIG Proceedings*. New Orleans, Louisiana, USA, November 16–20.

Leigh, J. & Johnson, A.E. (1996). Supporting Transcontinental Collaborative Work in Persistent Virtual Environments. *IEEE Computer Graphics and Applications*, 16(4), 47- 51.

Leigh, J., Johnson A.E. & DeFanti, T.A. (1996). CALVIN: an Immersimedia Design Environment Utilizing Heterogeneous Perspectives". *IEEE International Conference on Multimedia Computing and Systems*, 20-23.

Li, F. & Lau, R. (2003). VSculpt: A Distributed Virtual Sculpting Environment for Collaborative Design. *IEEE Transactions on Multimedia*, Vol. 5, no. 4, December 2003.

Li, H., Yin, G. & Fu J. (2006). Research on the collaborative virtual products development based on Web and X3D. *16th International Conference on Artificial Reality and Telexistence (ICAT '06)*, pp. 141-144.

Liesenborgs, J. (2000). *Voice over IP in networked virtual environments*. B.A. Thesis, School for Knowledge Technology, Limburgs Universitair Centrum, Belgium, 2000. Retrieved September 10, 2009, from http://research.edm.uhasselt.be/~jori/page/index.php?n=CS.Thesis.

Macedonia, M.R., Zyda, M.J., Pratt, D., Brutzman, R., Donald, P. & Barham, P.T. (1995). Exploiting reality with multicast groups: A network architecture for large-scale virtual environments. *IEEE Virtual Reality Annual International Symposium (VRAIS'95)*, North Carolina,

Miller, C. & Thorpe A. (1995). SIMNET: The advent of simulator networking. Proc. IEEE 83, 8 (Aug.), pp. 1114-1123.

Pandzic I., Capin T., Magnenat-Thalmann N. & Thalman D. (1996). Towards Natural Communication in Networked Collaborative Virtual Environments. In Proc. of FIVE '96, Pisa, Italy.

Pandzic I., Magnenat-Thalmann N., Thalman D. (1998). Realistic Avatars and Autonomous Virtual Humans. In R. Earnshaw & J. Vince (Eds.): *VLNET*

Networked Virtual Environments. Virtual Worlds in the Internet. IEEE Computer Society Press.

Redfern, S. & Galway, N. (2002). Collaborative Virtual Environments to Support Communication and Community in Internet-Based Distance Education. *Journal of Information Technology Education (JITE)*, Volume 1 Number 3, pp. 201-211.

RFC3261 (2002). *SIP: Session Initiation Protocol*. Retrieved September 10, 2009, from http://www.ietf.org/rfc/rfc3261.txt

RFC3550 (2003). *RTP: A Transport Protocol for Real-Time Applications*. Retrieved September 10, 2009, from http://www.ietf.org/rfc/rfc3550.txt

Schafer W., & Bowman, D. (2005). Integrating 2D and 3D views for spatial collaboration. *The 2005 international ACM SIGGROUP conference on Supporting group work*, pp. 41 – 50.

Snowdon, D., Churchill E. & Munro A. (2001). Collaborative Virtual Environments: Digital Spaces and Places for CSCW: An Introduction. In D. Snowdon, E. Churchill & A. Munro (Eds.) *Collaborative Virtual Environments: Digital Places and Spaces for Interaction*, ISBN 1-85233-244-1, Springer-Verlag.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(29):257-285.

Web3D (2009). *X3D and Related Specifications*. Web3D Consortium. Retrieved September 10, 2009 from http://www.web3d.org/x3d/specifications/.

Weber, J. & Parisi, T. (2007). An Open Protocol for Wide-area Multi-user X3D. *Web3D 2007 Symposium*, April 2007 University of Perugia, Umbria, Italy.

Wray, M. & Hawkes, R. (1998). Distributed virtual environments and VRML: an event-based architecture. *Comput. Netw. ISDN Syst*. 30, 1-7 (Apr. 1998), 43-51. DOI= http://dx.doi.org/10.1016/S0169-7552(98)00022-1

Yamazaki, Y. & Herder, J. (2000). Exploring spatial audio conferencing functionality in multiuser virtual environments (poster session). *Third international Conference on Collaborative Virtual Environments*, San Francisco, California, United States. Churchill, E. and Reddy, M. (Ed.) CVE '00. ACM, New York, NY, 207-208. DOI= http://doi.acm.org/10.1145/351006.351051.

Tsiatsos Th., Konstantinidis A., and Pomportsis A (2010). Evaluation Framework for Collaborative Educational Virtual Environments. *Journal of Educational Technology & Society*, Vol. 13, Issue 2, 2010, pp. 65-77.