A framework to generate 3D learning experiences

Ugo Barchetti, Alberto Bucciero, Stefano Santo Sabato
and Luca Mainetti
University of Salento
Italy

1. Introduction

A Collaborative Virtual Environment (CVE) is a computer-based virtual space that supports collaborative work and social interplay. In a 3D CVE, a 'hosting' 3D world is the necessary ingredient: within it users provided with graphical embodiments called avatars that convey their identity (presence, location, movement etc.), can meet and interact with other users, with agents or with virtual objects.
Even if graphics hardware and 3D technologies are rapidly evolving and the increased Internet connection speed allows the sharing of amounts of data and information among geographically distributed users, the development of networked three-dimensional applications is still complicated and requires expert knowledge. Although some collaborative 3D Web technologies and applications have already been developed, most of them are particularly concerned with offering a high level realistic representation of the virtual world since increasing the level of detail increases the sense of 'virtual presence' in the 3D world. However, these developments have not, at the same time supported a high level, non-expert authoring process and the concepts of programming flexibility and component re-use have rarely been taken into account.
In this introduction, we discuss our research experience in the field of Collaborative Virtual Environments. We will outline our approach which has been based on both multi-channel integration and on high performances issues.
Moreover, we advocate the need for a drastic simplification of authoring and personalisation phases. We propose this should be enacted through formal description of the sets of interactions, as well as of their behavioural features and rules, that we together call 'collaborative metaphors'. This should be done in a component oriented fashion to drive collaboration among users according to the designer's specifications. As result of previous considerations, we present OpenWebTalk (OWT), a declarative 3D component framework based on XML documents describing not only the formal structure of the environment of the virtual world, where the action takes place, but also the complex set of interaction rules that govern interactions between users and world objects used to stimulate certain kinds of collaboration. Such a framework would thus effectively help fast prototyping and an easy building up of collaborative applications. The OWT framework also provides a high

performance runtime 3D rendering engine, fully configurable through XML, in order to easily configure virtual world settings as well as collaborative interaction rules, thus allowing geometries, behaviours and content to be independently controlled, assigning different tasks to different developers (i.e. software dev. , content dev. , graphics dev. , session designers).

The main innovative features of the proposed framework are:

- Decoupling of all phases of the authoring process.
- Easy definition and composition of virtual sessions in a component-oriented fashion.
- Drive and control interaction among users to stimulate specific kinds of collaboration.
- High quality and performance rendering with support for post-processing effects.
- Easy configuration of specific 3D effects (such us 3d audio, camera effects, etc).

Moreover, even thought Collaborative Virtual Environments are one of the most successful applications in the field of Computer Supported Collaborative Work (CSCW), their support and working environment has often been relegated to desktop systems without regard for the recent spread of smart mobile devices and wireless networks.

In this work the OpenWebTalk framework is also presented as a means to generate 3D learning experiences. We explore the framework's capabilities of configuring itself on the basis of an XML meta-model declaration language representing not only the workspace, but also interaction rules shared among the 3D users during the collaborative experience. OpenWebTalk's photo-realistic rendering capabilities will allow us to use this architecture across a large range of applications that span from e-learning to architectural virtual reconstructions.

First of all, we explain our motivation for this definition of a 3D collaborative and configurable engine (Section 2). Then we describe the state of the art from both the scientific and the commercial point of view (Section 3). In section 4 we describe the history of OWT and our previous approaches (Section 4) and in section 5 we describe the OWT framework and its features and functionalities.

2. Motivations

In our previous research work we have collaborated in the design and development of CVE engines to be used in e-learning. In these earlier pieces of research the maintenance of lower system requisites to enable all kind of users to take advantage of collaborative experiences was crucial. Moreover, after scouting all applications and systems on the market, we identified two main problems:

- A number of the proposed solutions were applications rather than flexible frameworks. They therefore required major reprogramming work in order to be used either in different fields or new situations.
- A lot of systems strengthened the graphic quality (aiming at a kind of photo-realism) or the reproduction of physical laws rather than improving collaborative features and/or interaction possibilities in order to open up more opportunities for users to engage with the development process.

It was therefore that we decided to design and develop a new framework based on the client/server paradigm in order to match the rendering of high performance with the possibility of creating totally configurable collaborative sessions. The configuration of the sessions is to be intended to be both from a 'geometrical' point of view (i.e. which objects are in the world, where the light sources are, which properties the different objects possess, where the cameras are and what configurations of them the scene is based on) and from a collaborative point of view (i.e. what are the interaction rules between users and objects defined for the virtual environment in a specific period of time?). We realised that at present collaborative frameworks marrying the state of the art rendering techniques with the possibility of configuring each aspect of collaborative experiences are lacking.

3. State of the art

We carried out an in depth investigation focussing on two specific aspects in order to analyse the state of the art :

- One related to articles produced by the scientific community (we analysed the ACM and IEEE online archives);
- One related to commercial systems.

Distributed Virtual Environments are the bridge between virtual worlds and collaborative virtual environments. One of the first 3D distributed environments was DIVE (Distributed Virtual Interactive Environment) (Frécon and Stenius, 1998), created in Sweden by SICS (Swedish Institute of Computer Science) in 1991. It provided representation by avatars, different browsing and interaction types and audio and video communication.

The DIVE environment consisted of a group of distributed processes that represented avatars or objects, which could modify the database that contained the representation of the world. The communication among these processes was based on multicast protocols and thus did not rely on a central server architecture. In fact, each user possessed a copy of the world in which they collaborated.

After DIVE, other environments like SPLINE (http://www.merl.com/projects/spline/), NPSNET (Oliveira et al., 2000) or MASSIVE (Greenhalgh et al., 2000) appeared. Many of them have successfully solved problems such as the structuring of the world and the zone partitioning, the state propagation, the coordination and the consistency and the realism. They also introduced concepts such as that of portals between worlds, to be able to move between them. Those collaborative environments offered richer user interactions by means of flexible collaboration tools. Nevertheless, the complexity of the domain has led to huge and intricate software systems that are difficult to extend and re-use, thereby precluding system interoperability and flexibility. To cope with these problems, the trend is to augment modularisation in the design of such infrastructures. Many systems are following a framework-based approach to augment modularisation and provide extension points to develop new applications on top of them.

In the domain that we are studying, we highlight two examples of framework-oriented platforms enabling the development of collaborative virtual applications. One example is Ants CSCW (García and Skarmeta, 2003); a generic multi-user collaborative framework. The ANTS system is a component-based application framework that simplifies the development of distributed collaborative components in Java language. It provides a client-side container for JavaBean components that hides the complexity of the middleware layers by providing a remote persistent property mechanism and a distributed event service. The key architectural decision to create the distributed event service on top of a publish/subscribe notification

system permits a very powerful and flexible approach. State propagation is efficiently accomplished by the decoupled notification system.
From the commercial point of view, IRRLICHT (http://irrlicht.sourceforge.net/) is an open source graphic engine optimised for rendering quality and graphic effects. It is able to render high resolution images using the most recent techniques (per-pixel and per-vertex shaders, culling, bump-mapping, etc.). There is no support for collaborative features or for external configuration to make collaborative experiences.
Ultima Online (http://www.ultimaonline.com) (often shortened UO) is a fantasy MMORG (Massive(ly) Multiplayer Online Role-Playing Game) developed by Origin Systems, a software house recently bought by EA games. UO client offers the player the possibility of playing either in 2D isometric graphics or in 3D isometric graphics with the optimised use of shaders. The game used servers who were the owners of Origin in Europe and America. The UO server systems are hardcoded; everything is specified in the source files of the emulators (managing of the behaviour of objects, skill definitions, NPC advanced artificial intelligence) with only a minor possibility of personalising via an external configuration file.
World of Warcraft (www.worldofwarcraft.com) is an MMORG fantasy videogame which is only playable through the Internet by paying a fee. It was developed by Blizzard Entertainment and was published in 2004. The game has three dimensional graphics which enable the player to enter its virtual environments. World of Warcraft is the most played MMORG in the world with over 9 million active enrolments. The system is able to provide rendering which takes advantage of the most recent accelerated graphics cards (optimisations such as 'per-pixel shading' and 'per-vertex shading' are used). It is impossible to configure the game session externally.
Second Life (www.secondlife.com) is probably the best known product. Developed by Linden Labs, it allows the users who are connected to interact and perfectly imitate the interactions which would happen between users in the real world. The system uses a renderer based on OpenGL but which is hardcoded inside the client. It is possible to configure a rather small number of environmental parameters through external configuration files but the operations are very difficult and slow. Moreover is not possible to configure interaction rules inside environments.

4. History of OpenWebtalk and previous approaches

Shared virtual environments as collaboration tools (CVEs) are mainly intended as a way to support collaboration among several users working on a common (virtual) scene (data model). Communication between instructors and trainees (simulation and training applications), sharing data (for visually supported discussions between scientists or decision-makers) (CSCW), support for innovative teaching-learning and support for collaborative e-learning (CSCL) are all examples of uses for shared virtual environments. These applications (re)create a multi-user virtual world (Damer, 1997), as two or three-dimensional graphical environments inhabited by users (represented as digital actors called 'avatars') that share time, space and actions with other users, cooperating together for a common goal.
Several different software systems today, both commercial and research prototypes, support Collaborative Virtual Environments. Our work started in 1998 with the development of

WebTalk, which has evolved, over the years, into the current OpenWebTalk, described in the next section.

4.1 WebTalk04

After several experimentations (http://www.seequmran.net/) with 3D educational applications, we have developed the WT04 framework (Barchetti et al., 2005). This development sprang from the need to create 3D collaborative learning experiences with significant differences (in terms of world geometries, interactive objects, textures, collaboration features, etc). For each different application different versions were needed (e.g. selecting different content and different quizzes for different groups of schools) to prepare specific experiences. Flexibility and configurability were revealed, therefore, to be a crucial requirement. Moreover, in order to expand the number of users, we needed to improve both performance and reliability whilst decreasing, if possible, requirements on the connection bandwidth. A number of key decisions were therefore taken for future development:

- The choice of a new supporting platform to be Macromedia Studio 2004 MX: the widespread availability of the Macromedia Shockwave player (counting over 200 million installations all over the world) and its 'industry reliability' were crucial elements. Some technical features were also relevant: e.g. the capability of importing 3D studio file format, the support for the Havok physics engine (http://www.havok.com/) (able to control and assign physics to both unanimated and animated objects and control their kinematics), the presence of a set of built in behaviours to control, for example, avatar movements.
- Use of XML as a configuration language: the possibility of 'describing' static features (e.g. representing position, colour, dimension etc. of the objects at start-up) and a dynamic features (e.g. defining whether objects can be moved, clicked etc.), instead of 'hard coding' them, was essential for gaining configurability and flexibility.
- Creation of a specialised parser to 'read' the configuration file: commands, written in an Xpath like syntax, are used in order to find attributes and metadata stored in the (XML) configuration file.

Moreover an XML representation is well suited to describing the 3D scene as structured data that can be processed without paying attention to how the data should be presented. The conversion style sheets in fact allow the switch from one format to another. Through different XSL files the content of a WT04 XML scene graph file can be easily converted to VRML or X3D, to pretty-printed HTML or to any number of other formats.

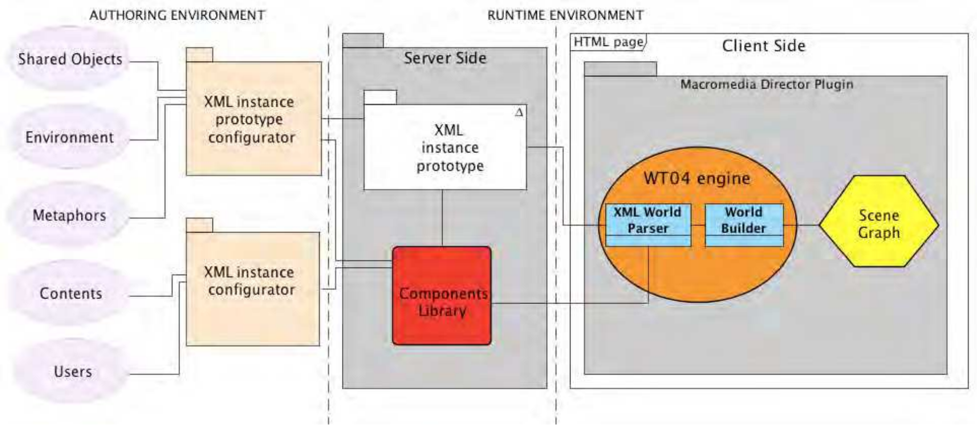

Fig. 1. The WebTalk04 architecture

Expression of scenes in XML enables the application of a wide range of existing and emerging XML-based tools for transformation, translation, and processing. XML provides numerous benefits for extensibility and componentisation, as well as the ability to develop well-formed and validated scene graphs; an extremely valuable constraint since 'broken' 3D content would no longer be allowed to escape onto the Web where if might cause larger scenes to fail.

Finally, a formal XML description of the scene graph and the user-to-user or user-to-object interactions seems the best 'interface' between the WT04 and the following two subsystems represented within the whole architecture as in Figure 1 :

- WT04 authoring environment where the designer defines and fine tunes the world structure and the specific behaviours attached to each user or interactive object.
- WT04 runtime environment, in which this XML file works as a declarative description of how the virtual world has to be rendered as well as of the ways in which the action can take place and evolve.

4.2 Mobile WebTalk

The development of CVE systems is now a well-established reality. A CVE (Collaborative Virtual Environment) is a virtual 3D place representation, in which users, being represented through graphical embodiments called avatars, share the same experience, working and interacting with the same set of virtual objects. In this case users see the same representation of shared workspace (i.e. a 3D world) and the effects on the users' or objects' actions are also the same for all. This paradigm is known as WYSIWIS ('what you see is what I see').

The availability of mobile location aware technologies on consumer devices has encouraged researchers to consider how to join the classic desktop CVE paradigm with a newer one based on mobile devices, in a mixed reality oriented fashion. In this case the paradigm is no longer 'WYSIWIS' as in a classic Collaborative Virtual Environment, but WYSINWIS ('what you see is NOT what I see'): users see different representations of the same workspace (both environment and objects). Such systems are called CMVRs (Collaborative Mobile Virtual Reality systems) in which avatars in the virtual world and mobile players in the real world share the same experience, aiming towards a common goal even if it is provided through

different devices, with different capabilities and therefore with different views of the same workspace.
MoWeT (Mobile WebTalk) (Bucciero et al., 2007) takes its form from these considerations, trying to extend the scope of a 'pure' CVE such as OpenWebtalk, towards a Mixed Reality System developed as a mobile, flexible and configurable framework. Even though MoWeT represents a part of a greater framework, in this sub-section we focus on the mobile subsystem, as WebTalk04 itself has already been the object of some papers (Barchetti et al., 2005) (Barchetti et al., 2006). Furthermore MoWeT can also be configured as standalone system, becoming, in this way, independent from any other part of the WebTalk04 framework. MoWeT is particularly oriented to the cultural and tourist context, for uses such as mobile interactive guidance for tourist experience, with extensive multimedia content support (video, sound, hypertext, etc).
The MoWeT framework is designed to provide a flexible and configurable solution that implements integration between RFID (Radio Frequency Identification) and GPS localisation technologies, allowing users to share the same experience in a collaborative environment. This collaboration is achieved through a client/server architecture. The client application is specifically conceived in a modular way in order to separate the specific tasks into different classes; it manages maps visualisation, connections (GPS, RFID, Internet) and event/action interactions. The server, by contrast, keeps track of user's movements and sends this information to other users.

5. OpenWebTalk

As already mentioned, in the last few years hardware technology, high bandwidth networks and 3D graphics systems have brought Virtual Reality (VR) technology into a new field: Collaborative Virtual Environments (CVE). The goal of the applications and systems in this domain is to use computers as tools for communication and information sharing through 3D environments working as shared workspaces that provide collaboration facilities for several branches of applications (simulation and training, e-learning, collaborative working, etc.). Historically, such systems have mostly focussed on the 'quality' of graphics (often intended as a kind of photo-realism), and 'real-world behaviour' (i.e. systematic replication of physics laws) rather than collaborative features and/or features that engage the user. As an improvement on these previous approaches, we advocate the need for a framework that combines high performance in real-time rendering with a huge grade of configurability. This would allow the fast prototyping of a large number of 3D collaborative environments in an easy way, without spending so much time reprogramming the system. Moreover the use of photo-realistic graphics should enhance the level of engagement during the collaborative experiences bestowing these virtual environments with a high sense of virtual presence. In this chapter we present OpenWebTalk (Barchetti et al., 2008), a framework to support high performance and high resolution 3D collaborative experiences that is based on the experiences we have already had with WebTalk04 and MobileWebTalk, to support, in a flexible way, high performance real-time collaboration.

5.1 Experience in the applied research projects

Nowadays the research and the development of technological solutions are continuously evolving. People from any part of the world are able to collaborate in order to reach common objectives. Typical examples of this are evolved systems of teleconferencing and distance learning. Most of such systems, called Computer Supported Collaborative Workgroups (CSCW), use computers as a means for sharing notions of space and time across very great distances. Each of them exploits a workspace that is a shared place where users can collaborate, acting locally and then spreading their actions towards the other users of the group via the defined informative infrastructure. Collaborative Virtual Environments (CVE) represent a particular type of CSCW system and use a workspace which exploits a three dimensional representation to increase the sensation of mutual awareness. In fact, CVEs emphasise the awareness of the workspace and of the reciprocal interactions with the other users who are present, as though they were in a virtual environment.

CVE systems are used in the simulation of environments in industrial or military fields, and also in collaborative games and collaborative e-learning. Recent research on CVE has shown that creating an 'appealing' virtual environment from a rendering point of view is not enough. It is also absolutely necessary to integrate it with interaction rules between users and the environment. In this way the engagement, seen as the users' satisfaction rate with their interaction with others in the virtual environment, can be improved. CVE engines need high computing resources therefore, in past years, the scientific community has chosen to properly tune the graphics rendering quality and the number of possible interactions in order to grant the execution also for less skilled clients. Another widespread peculiarity of CVE engines is that the interaction rules, the geometry and all the necessary settings on the client during the execution are hardcoded inside the client application; this is to be downloaded by users before taking part in the collaborative session. We will show the importance of matching high quality rendering with the possibility of configuring in a simple manner not only the structure of virtual worlds, their objects, the textures that cover them, the light effects, the position and orientation of cameras, but also the interaction rules in environments. This is the way to speed up and make economically profitable the creation of streams of collaborative sessions starting from a template or a prototype without lowering the high quality of rendering. In the last years, we have proposed and developed a framework called WebTalk04 (WT04) able to generate and allow the fruition of educational collaborative sessions. This was obtained by giving the users the possibility of connecting to the 3D environments through the web browser using a low-resolution renderer (using the Adobe Shockwave player plug-in) and using poor quality graphic performance. Thanks to this framework throughout the years, hundreds of sessions in projects such as Learning@Europe (www.learningateurope.net), Storia@Lombardia (www.storiadella lombardia.it) and Learning@SocialSports (www.learningatsocialsport.it) have been given to thousands of users worldwide. These projects have shown the necessity of generating streams of sessions in a simple and fast way, so that they start from a common template and will only differ in the specific content of each single session while still maintaining high graphic performance measured in terms of rendering speed and overall quality of generated images.

5.2 OpenWebTalk Framework Architecture

The OpenWebTalk architecture is based upon a client/server paradigm and exploits an application server hosting a Web Server for static and dynamic contents and a server-side Daemon (the OpenWebTalk server communication manager) for sharing data in cooperative and distributed applications. While the server is listening to client connections and distributing events between the participants, maintaining a centralised state repository, the client-side is composed of an application which runs both the rendering and the collaboration modules. The collaboration logic, via the event and behaviour manager, detects every event generated in the 3D interface and forwards them, through the OpenWebTalk Communication Protocol, to the server. Likewise, any other events generated at the same moment by other clients connected to the same environment, are distributed by the Server Communication Manager to the Client's Communication Manager on client-side, which manages them by running the right action. Users can thus see each other's avatars and objects moving and operating in response to the events and to the 3D world solicitation, collaborating with each other during the online sessions. All the system architecture has been developed using C++.NET 2005, built up around the MVC (Model View Controller) design pattern in which the model represents the shared world state, the view represents each client application, and the controller is the programming logic that creates and regulates each end-user GUI (Graphical User Interface) as well as the shared environment. The architecture of OpenWebTalk is shown in Figure 2. Let us now describe all its main components (a description of each module will be given later):

Client-side

The 'world engine' is the client-side of the application. Through a number of modules, it manages the connection with the server and controls the rendering on the client machine. The 'renderer' and 'collection' modules, based on the G3D (http://g3d-cpp.sourceforge.net/) rendering framework, render in real-time the scene according to the shared state of all the objects and avatars. G3D is a commercial-grade 3D Engine available as Open Source (BSD License). It makes low-level libraries like OpenGL and sockets easier to use without limiting functionality or performance (G3D is entirely written in C++). Due to performance issues we implemented our framework using the low-level libraries of G3D calling directly the API functions and just re-writing some classes to improve the overall performance.

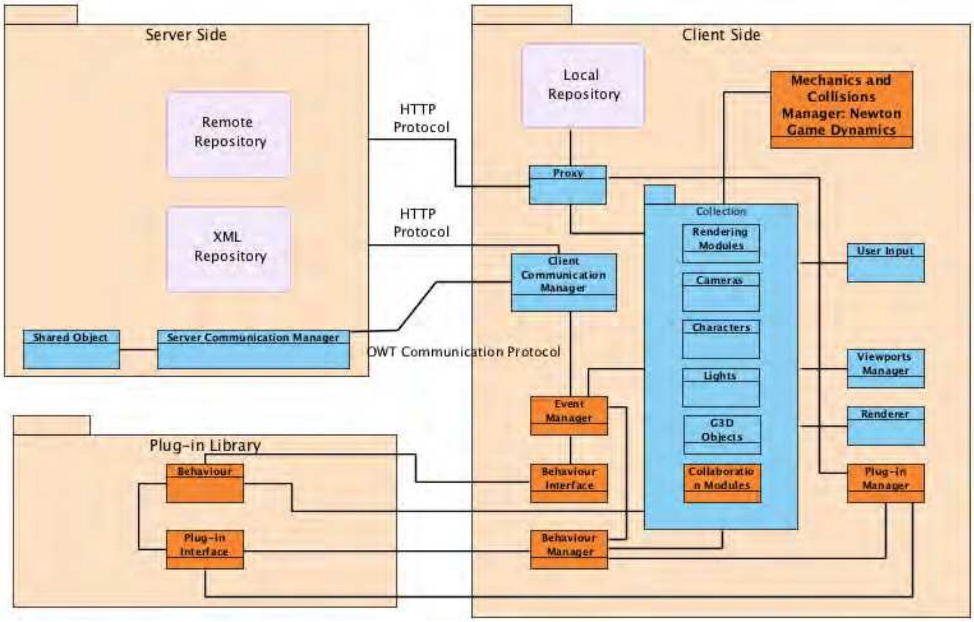

Fig. 2. The OpenWebTalk Architecture

The ‘collision module’ computes in real-time the collisions between the entities in the virtual world introducing physics into the virtual environment. The proxy module manages the object downloading and caching features, using a smart optimised algorithm, it creates the right download list after verifying that there is not a local copy of the object in the local cache system. The ‘interaction engine’ is composed of event and behaviour managers that control the collaboration features. We show in detail what this is and the way in which it works in the section called ‘OpenWebTalk Configuration and Collaboration Model’.

Server Side

The *Contents Repository*: stores all the geometries and the textures, created at the design stage.

XML file repository: this stores all the different configuration (XML) files. When a client starts a session, the system selects the right XML file and loads it.

Communication Server: this traces the actual shared state, keeping track of, for example, positions, rotations, and ‘state’ in general of all the avatars and objects. Any shared state change is sent to all the clients who update the environment state. In order to deploy the OpenWebTalk framework architecture, we can build it both for Microsoft Windows based systems and for UNIX like ones.

Figure 2 is a detailed schema of the main modules that compose the OWT architecture.

5.3 OpenWebtalk Features and Functionalities

In this section we describe the features and functionalities that characterise our framework. We also describe our implementing choices to generate collaborative experiences based on well-defined set of rules.

Rendering

As said before, rendering performance and optimisation are very important for a real-time 3D collaborative framework. We decided to customise the G3D rendering module to adapt it to our framework. Thanks to the G3D performance and capabilities, we can now render full-screen virtual environments in real-time, using optimisations like shaders (Regenbrecht et al., 2006), shadow map and shadow volume (Pettifer and Marsh, 2001) programming. Moreover we can control the hardware acceleration performances at run time so as to optimise the frame quality. In the render module we introduced optimisation that can be grouped into two different classes:

- Image quality optimisations
- Performances optimisations

In the image quality optimisation class, we grouped all the optimisations that concern the rendered-frame quality from a qualitative point of view: the render module dynamically selects the right mix of optimisations to dynamically balance quality with performance. Our renderer module uses bump mapping, per-pixel and per-vertex rendering optimisation.

Level of detail and Frustum culling optimisation are used during the performance optimisations to improve the rendering speed (measured in Frames per Second [FPS]). Our render module (based on the G3D render module) selects the right level of detail based not only on the standard parameters such as distance and visibility, but also on the overall performance of the system.

G3D (http://g3d-cpp.sourceforge.net/) is also a commercial-grade C++ 3D engine available as Open Source (BSD License). It is used in commercial games, research papers, military simulators, and university courses. G3D supports real-time rendering, off-line rendering such as ray tracing, and general purpose computation on GPUs and is the rendering engine we used in our research work. G3D provides a set of routines and structures so common that they are needed in almost every graphics program. It makes low-level libraries like OpenGL and sockets easier to use without limiting functionality or performance. G3D is a rock solid, highly optimised base on which to build a 3D application.

The Physics engine

The physics engine is very important in applications in which 'virtual presence' is a key feature. In fact it is crucial to have a physics engine that can reproduce all the physics interactions between users and objects in the virtual environment. The OWT physics engine is based on an open source module that calculates Mechanics and collision inside the environments. This module is Newton Dynamics (www.newtondynamics.com). Newton Game Dynamics is an integrated solution for real-time simulation of physics environments. It offers API that provides scene management, collision detection and dynamic behaviour. Our engine implements a deterministic solver which is not based on traditional LCP or iterative methods but possesses the stability and speed of both respectively. This feature makes our solution a tool not only for games, but also for any real-time physics simulation.

Newton Game Dynamics can be integrated into our project with ease. With this technology the user only needs to know basic physics principles to produce realistic physics behaviour.

Configuration and Collaboration Model

To bestow our virtual environment with a high sense of virtual presence, in addition to the full graphic driven paradigm, the overall application takes into account and provides a set of interaction features to drive the collaboration among users in the specific way the designer intends it to do so. OpenWebTalk proposes a declarative format to reduce the need for programming for non-expert designers of collaborative experiences. For this reason we examined the major issues concerned in the formal description of a collaborative virtual environment both from a static point of view; regarding how the virtual world is when it is generated, and from a dynamic point of view regarding how it can evolve during the collaborative sessions through a declarative description of the Event Conditioned Actions (ECA) that can be performed in it. We also manage 'collaborative metaphors' that we can define as a set of basic rules which describe different modalities of interaction between users and between the users and the environment (Barbieri and Paolini, 2001). These rules encompass various aspects, such as the way users can gather in groups, the possible ways of chatting to each other, the way to navigate in the virtual space, how to visualise the world and objects, etc.. Taking decisions on all these aspects shapes a collaborative situation.

In figure 3 we can see the OpenWebTalk declarative system levels which are composed of:

1. Scene graph and behaviour schema: coded through XML-Schema it represents the structure of a valid OpenWebTalk compliant XML-instance, where the elements composing XML-instances, hierarchical rules between elements and supported data types, have been defined.
2. Scene graph and behaviour prototype: a XML document representing the skeleton of a collaborative session where the designer has already fixed geometries forming the virtual environment as well as the main user interaction rules.
3. Scene graph and behaviour instance: the final and XML instance completed with all information and content depending on the specific context (specific users involved, specific target of the experience, particular topics given).

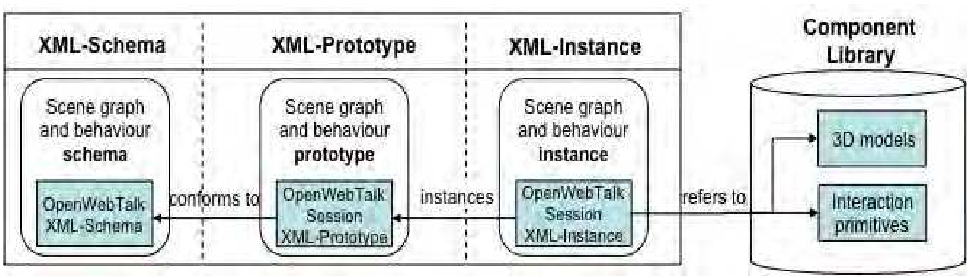

Fig. 3. OpenWebTalk declarative system levels

When an XML instance is generated, the system is ready to start a collaborative session. The XML file is sent (at run time) to the clients' runtime environment which interprets the declarations and provides the instance with the right components taken from a library (on

the one hand, it instances the right 3D models in the world, and on the other, it sets up the collaboration rules that will govern the shared experience).
The OpenWebTalk scene graph is organised as a sequence of 3D environments called parts in which users can navigate, moving from one to another by simply colliding with special interactive objects, often in the form of ports or gates working as teleports, thus causing the unloading of the current part rendered by the engine and the loading the next one.
Looking at figure 4, after a configuration node that specifies some important first-start configurations such as the collaborative server IP address, the world description node is divided in <part> elements, each describing a single environment in which the session takes place composed of the <avatar> element and a list of <objectG3D> elements. Each part is characterised by its environmental container, a geometrical file that is considered as a 'stage' in which we instance avatars and objects. In the avatar section we put the geometrical peculiarities and all the attributes needed. Each objectG3D section is composed of a <geometry> element (describing the geometry model, position, rotation etc), and a <behaviour> element (describing the possible interaction rules).

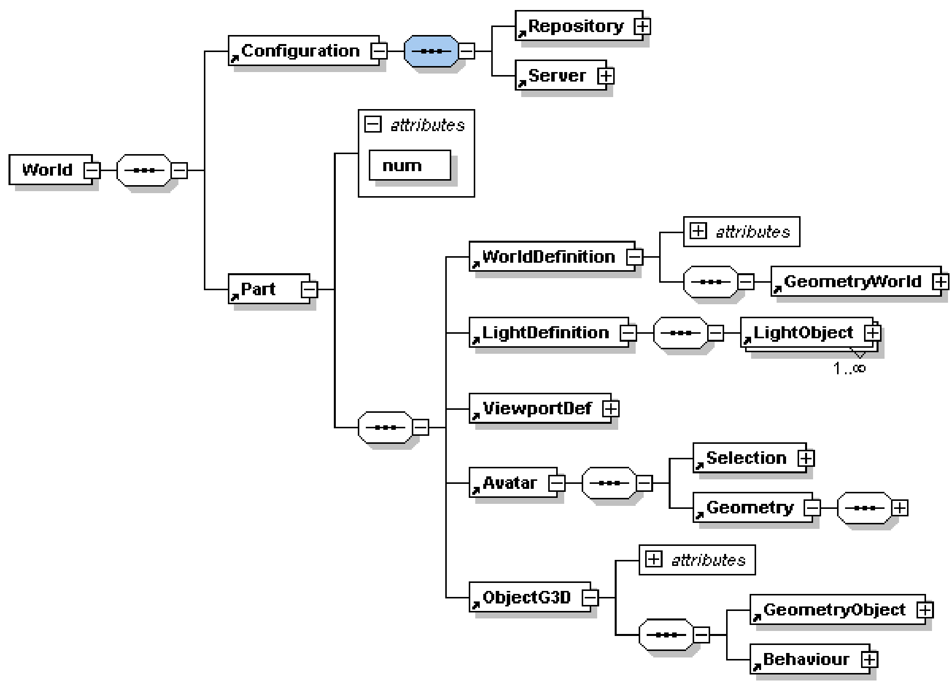

Fig. 4. Configuration XML Schema

In OpenWebtalk, user-to-user or user-to-object collaboration issues are intrinsically correlated to interaction primitives that we call virtual actions supported by the environment; these primitives are able to control and drive collaboration in a specific and well-defined scripting language.

In our model, interaction is achieved and controlled through a mechanism based on three main concepts using the paradigm of Event Conditioned Actions (Figure 5):

- **Entity**: a general resource within the system. An entity may be a shared object with a graphical representation, or an abstract concept with no visible representation such as, for example, a server side remote shared object mapping a certain property of the system. Such entities provide a set of properties representing in some way the inner state of the entity and an amount of listeners to which the entity is registered, that allow events to react and perform one or more actions. In this view users and their own avatars can be considered a particular kind of entity
- **Event**: is a trigger that can be raised by some user interaction or by the system itself, notifying that something has happened in the shared environment.
- **Action**: actions are 'atomic instructions' that are invoked in a particular time instant by someone (typically an avatar) or something (typically an object). So actions are modules that we can configure in order to obtain the dynamic reaction we want.

According to the classical view, the Event Conditioned Actions' rules are based on the following form:

$$\begin{array}{r} \text{on event} \\ \text{if conditions} \\ \text{do actions} \end{array} \tag{1}$$

Moreover, in our declarative model the central concepts around which the whole scene graph is described are the objects and users or, in more general worlds, entities.
Then we derived the following entity-oriented interaction control model.

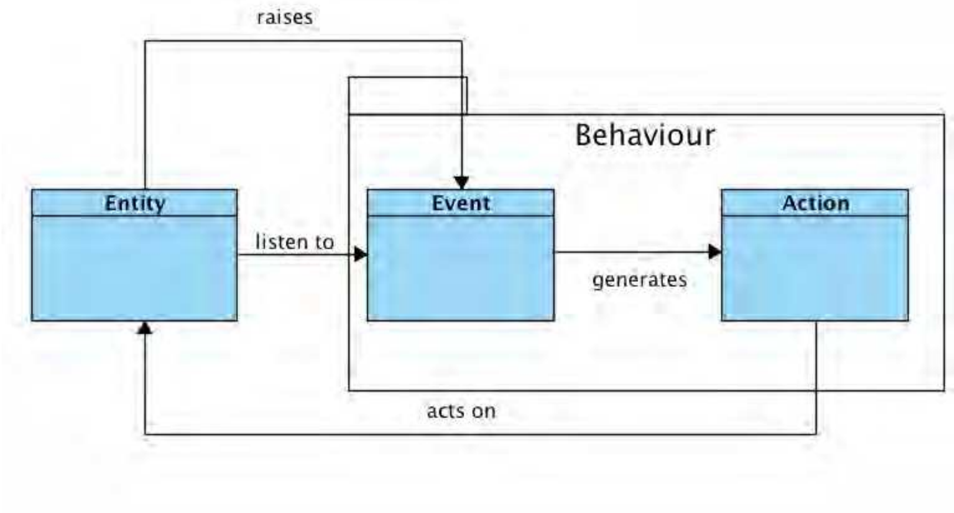

Fig. 5. Entity-Event-Action paradigm

In this model (detailed in figure 5) the main component is entity that can be thought of as an abstraction of both users as active actors of the system who are allowed to raise events, and objects which are passive components of the system which listen to events propagated within the virtual environment and react to them by performing one or more actions. The

target of the action can equally well be the same entity which caused in some way the event, or another, different, one.
The group made up of event and action takes the name 'behaviour' as it states the manner in which some entities evolve the system state when perturbed.

3D Positional Audio

In this subsection we will speak about the 3D positional audio feature. This feature manages the audio rendering into the 3D collaborative environments. The 3D audio is very important for improving the 'virtual presence' during the sessions. Every sound source is rendered in its 'real' position inside the virtual word, reproducing the sound that a real user would have heard in the real world. Our objective was to reproduce 'real' audio feelings in the virtual worlds. More in depth, we want to generate:

- **Environmental Audio**: this kind of audio source will increase the 'virtual presence' inside the 3D world, it will be loaded and executed when the framework is loading the virtual world.
- **Avatar Audio**: this reproduces all audio produced by the avatar's movements inside the virtual worlds.
- **Audio Sources into the 3D parts**: different and independent sources are included; they have the physical attributes and they can be listened within the virtual world. Each source will be customised and independently and individually managed by the user through different modes of control.

All the audio sources are 'positional', so they are closely dependent on the position, the direction and speed of movement of the character.
A great many techniques for reproducing sound in different ways have been developed over the last two decades but only three of these have been named: mono, stereo and quadraphonic, positional audio.

- **Mono**: The first audio rendering technique developed was mono, the simplest technique for reproduction of sound development and one that also offers fewer benefits. The mono is a single information flow noise to be emitted from a single loudspeaker positioned facing the listener. For special requirements, such as the need to serve a wide area, that information flow can also be played through several speakers, but the information reproduced by the different speakers will always be the same.
- **Stereo**: Then, in order to give more space to the sound than that permitted by mono and thereby make it more realistic (sound in nature almost always originates from multiple spatial points), stereo is a technique for reproducing sound in a more complex manner. In the stereo there are two acoustic information flows, each of which is intended to be spread by a different loudspeaker positioned facing the listener, one on the left and one on the right, under fixed rules. The sound that implements the stereo sound is called 'stereo' or, more briefly, 'stereo audio'. In stereo audio there are two information streams representing the two electronic information flows.
- **Quadraphonic (multi-channel)**: The stereo audio can be also multi-channel audio; here the audio shows more electronic information flows each of which represent a different information flow. These flows are called electronic channels of audio, and stereo audio presents two. The multi-channel audio will be used later to implement

all the techniques for reproducing sound developed since the stereo using more acoustic information flows. Extending this concept it is possible to implement multi-channel systems in which it is possible to have audio flows each of which represents an information channel.

- **Positional Audio** : 3D simulation is the most advanced group of audio3D effects. Using the 'head-related transfer functions' and reverberation, the sound changes along the path from the source (including the reflection on walls and doors) to the ears can be completely simulated in a more than realistic way. These effects include the location of sound behind, above and below the listener.

As said we tried to fully integrate the management of 3D sound in the framework OpenWebTalk. This question could have been resolved in various ways and because of these options a careful analysis of already existing 3D collaborative environments was made in order to understand how others have tried to resolve this difficulty, to investigate all the resources, open-source or not, that current technology makes available to the developer. We analysed OpenAL (http://connect.creativelabs.com/openal/default.aspx) (Open Audio Library) which is a multi-platform audio API free software. It is designed for efficient rendering of the positional audio in three dimensions. Its API style and conventions deliberately follow those of OpenGL. Definitely the most comprehensive library of all, and perhaps the only comparable one, is irrKlang (http://ambiera.com/irrklang/) which unfortunately gives way in the case of the need for high performance. The choice to use OpenAL was based on three key items: Buffers, Sources and Listeners (Figure 6). A buffer can be filled with audio data, and can be attached to a source. Each source contains a pointer to a buffer; the speed, position and direction of a sound and its intensity.

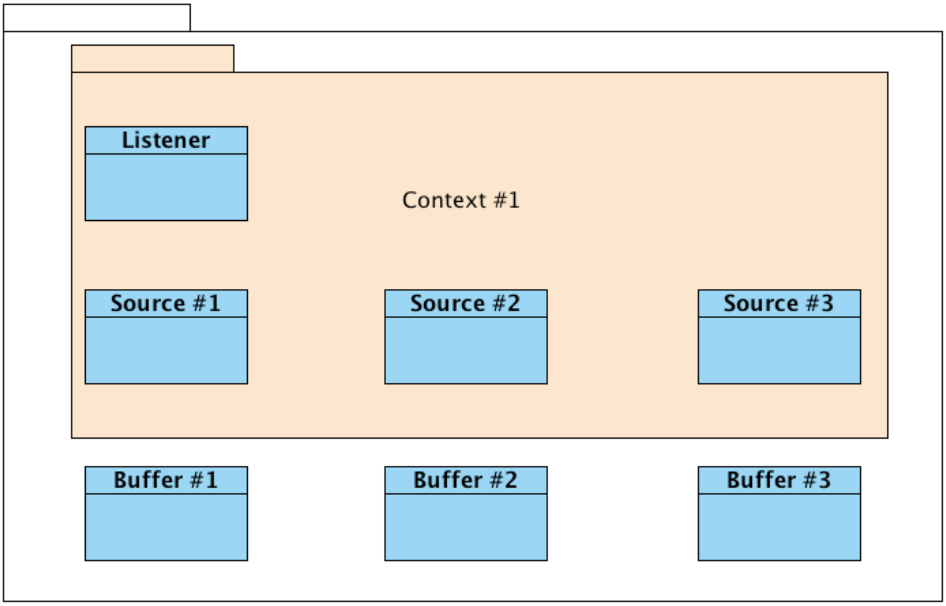

Fig. 6. The OpenAL architectural model

The source can be placed and left to play. The way in which that source is heard is determined by its location and its orientation relative to the Listener (there is only one Listener object) that contains the speed, position and orientation of the listener and the overall gain is applied to all sounds .
The buffer contains the audio data in PCM format, which is 8 to 16 bits, in both mono and stereo. The rendering engine performs all the necessary calculations as well as the attenuation due to distance, Doppler Effect, etc.
It is simple to create a number of sources and a single buffer and Listener and update the positions and orientations of the sources of the Listener and thereby dynamically manage to simulate a world with decidedly convincing 3D audio.
In order to improve 'virtual presence' it is very important to create audio modules capable of reproducing the sound flows inside the virtual environments. Let us examine how the OWT architecture is changed after the integration of the audio modules. Three key elements were incorporated in the GLOBAL COLLECTION and they are SPEAKER, LISTENER and AUDIO BUFFER; all three elements are derived from SOUND DEVICE, parent of all changes of the audio architecture. An extension called AUDIO ENVIRONMENT was also added that is used from the speaker if it is necessary to have some more specific features.
The following presents a more in-depth description of the single modules:

- **Sound Device**: is the 'father' of all that was added within OpenWebTalk to do with the sound; without it, it would not be possible to listen to any audio source in the application.
- **Listener**: without this there could be no definition of the 3D positional audio. It is placed (mounted) on the avatar and it depends on what the user hears, and especially how they 'listen' to it.
- **Audio Buffer**: these are the elements the speakers are filled with, and allow the speaker to play. Each buffer can be shared between multiple speakers.
- **Speaker**: undisputed centrepiece and main element in the case of audio OpenWebTalk; the speaker is the element with the most functionality of all.
- **Audio Environment**: expands the potential of the speaker regarding the most crucial and complex aspects of sound positioning such as reverb or Doppler effect.

All these items added to the architecture meet the basic rule of the OWT framework; they are loaded via XML and are shared between all the users of that particular world at that particular time.
We have also implemented a plug-in for the sound management of behaviours as a result of some BEHAVIOUR established by xml to specified objects.

6. Conclusion

This chapter shows the evolution of our research that brought us to the development of the OpenWebTalk framework. We presented the OWT framework emphasising rendering performance features, configurability and flexibility according to our configuration and collaboration model. Our framework used G3D as the rendering engine, one of the most important rendering engines in the commercial and scientific arena. It uses hardware acceleration to grant high performance and high rendering accuracy. We used 'Newton dynamics' as the external physics module that allowed us to introduce into the framework very accurate collision and occlusion calculation without overloading the framework.

Moreover, with the audio module we introduced a rendering multi-channel positional audio into the 3D environment to emphasise 'virtual presence'. Then we highlighted the configurability and flexibility aspects of the OWT framework using our configuration and collaboration model where it is possible to define sets of rules (behaviours and metaphors) according to the Entity-Event-Action paradigm. This approach can be useful to generate 3D learning experiences and to easily customise previous experiences. This framework has been successfully used in Social-Culture project (http://scultureproject.unisalento.it/) that aims to emphasise cultural heritage through collaborative virtual navigations (virtual visits) of three-dimensional archaeological museums.

7. Acknowledgment

We would like to thank the HOC Laboratory of Politecnico di Milano, and in particular Prof. Paolini and his staff, for their help in the identification of the issues faced by the learning projects such as learning@europe and learning@socialsport experiences.

8. References

Barbieri, T. & Paolini P. (2001). Co-operation Metaphors for Virtual Museums. *Proceedings of Museums & Web 2001*. April 2001, Boston, USA.

Barchetti U. ; Bucciero A. ; Mainetti L. & Santo Sabato S. (2005). WebTalk04: a Declarative Approach to Generate 3D Collaborative Environments. *Proceedings of the 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*. ACM, 55–60, November 2005, Pisa.

Barchetti U.; Bucciero A.; Mainetti L.; Santo Sabato S. (2006). WebTalk04: A Framework To Support 3D Collaborative E-Learning. *Proceedings of The Fifth IASTED International Conference on Web-Based Education (WBE2006)*. 13-18, January 2006, Puerto Vallarta, Mexico.

Barchetti U. ; Bucciero A. ; Mainetti L. & Santo Sabato S. (2008). OpenWebTalk a framework to support high performances and high resolution 3D collaborative experience. *Proceedings of the International Technology, Education and Development Conference*, March 2008, Valencia.

Bucciero A.; Barchetti U.; Mainetti L. & Santo Sabato S., (2007). Mobile Webtalk A Framework to Support Ubiquitous Collaboration. *Advanced Technology For Learning*, Vol. 4, 106-117, 1710-2251.

Damer, B. (1997). *Avatars! : Exploring and Building Virtual Worlds on the Internet*, PeachPit Press, Berkeley, USA, 0201688409.

Frécon, E. & Stenius, M. (1998). Dive: A scaleable network architecture for distributed virtual environments. *Distributed Systems Engineering Journal*, Vol. 5, No. 3, 91—100.

García, P. & Skarmeta A. (2003). ANTS Framework for Cooperative Work Environments. *IEEE Computer Magazine*, Vol. 36, Issue No. 3, 56-62.

Greenhalgh, C. ; Purbrick J & Snowdon D. (2000). Inside MASSIVE-3: Flexible Support for Data Consistency and World Structuring. *Proceedings of the CVE 2000 Symposium*. ACM, 139-146, September 2000, San Francisco.

Oliveira, M. ; Crowcroft, J. & Slater, M. (2000). Component Framework Infrastructure for Virtual Environments. *Proceedings of the CVE 2000 Symposium*. ACM, September 2000, 139-146, San Francisco.

Pettifer, S. & Marsh J. (2001). A Collaborative Access Model for Shared Virtual Environments. *Proceedings of the Tenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, June 2001, 257, Cambridge (USA).

Regenbrecht, H.; Haller, M.; Hauber, J. & Billinghurst, M. (2006). Carpeno: interfacing remote collaborative virtual environments with table-top interaction. *Virtual Reality*, October 2006, Vol. 10, No. 2, 95-107, 1359-4338.

New Achievements in Technology Education and Development

Edited by Safeeullah Soomro

ISBN 978-953-307-066-7

Hard cover, 460 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

Since many decades Education Science and Technology has an achieved tremendous recognition and has been applied to variety of disciplines, mainly Curriculum development, methodology to develop e-learning systems and education management. Many efforts have been taken to improve knowledge of students, researchers, educationists in the field of computer science and engineering. Still many problems to increase their knowledge on daily basis so this book provides newly innovations and ideas in the field of computer science and engineering to face the new challenges of current and future centuries. Basically this book open platform for creative discussion for future and current technologies to adapt new challenges in education sector at different levels which are essential to understand for the students, researchers, academic personals and industry related people to enhance their capabilities to capture new ideas and provides valuable contribution to an international community.