OpenWebTalk: a 3D high-performance Framework to support collaborative learning.

Ugo Barchetti*, Alberto Bucciero*, Antonio Capodieci*, Luciano Capone*, Luca Mainetti*,

Stefano Santo Sabato*

* University of Salento/ GSA-Lab[1] Department of Innovation Engineering, Lecce, Italy

Index Terms: eLearning, Collaborative Learning, CVE, WYSIWIS, WYSINWIS, CSCW.

Nowadays the term “eLearning” is used interchangeably in so many contexts that it is critical to be clear what one means when one speaks of it, and it is gaining increasing momentum as a teaching approach, both for education and training. In the field of e-learning we focalize our research on the “collaborative learning” paradigm. In this paper, we advocate the necessity to intermingle the present e-Learning approach, mainly based on bi-dimensional multimedia presentation of the contents, with new types of eLearning, based on the exploitation of high-defined 3D representation, thus adding one dimension to the current way of thinking eLearning.

To this end, we chose to start our research creating a 3D CVE system, based on a rendering engine borrowed by the serious game world.

This class of engines brings up high-definition rendering features with an high engagement level during the game experience. To use this kind of applications in eLearning scenarios, we need to add specific multi-user capabilities (to enhance user-user collaboration) and the possibility to simply configure the learning session. To this end we present OpenWebTalk, a framework to support high performances and high resolution 3D collaborative experiences, that is based on the experience we have done with WebTalk[2][3], a 3D CVE system mainly used for collaborative educational aims, to support, in a flexible way, high performances real-time collaboration.

The paper will show the framework’s capabilities to configure itself on the basis of an XML meta-model declaration language representing not only workspace, but also interaction rules shared within the 3D users during the collaborative experiences. This would allow the fast prototyping of a large number of 3D learning session in an easy way without spending much time in reprogramming the system. Moreover the use of photo-realistic graphics should enhance the level of engagement during the collaborative learning experiences conveying these virtual environments a high sense of virtual presence.

The OpenWebTalk photo-realistic rendering capabilities will allow us to use this architecture in a large range of eLearning applications. After the in-detail framework description, we present the results of an early experimentation performed by two teams of students.

This results will show how the framework’s performances scale with the scene complexity, and with the number of users concurrently connected.

Motivations

Nowadays we have a continuous evolution of the research and of the development of technological solutions which enable people from any part of the world to collaborate in order to reach common objectives. Typical examples of that are evolved systems of teleconference and distance learning. Most of such systems called Computer Supported Collaborative Workgroup (CSCW) [1], use computers as means for sharing the notions of space and time also at a very long distance. 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) [2] represent a particular type of systems CSCW and use workspace which exploits a three dimensional representation to increase the sensation of mutual awareness. In fact they emphasize 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.

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

In the CVE systems, each user is represented, in the environmental container, a kind of stage were the interactions happen, by an avatar, a kind of figurine, to avatars of other users in order to respect some interaction rules. The avatar operates in the virtual world interacting with both objects and other avatars.

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 a virtual "appealing" 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 in their interaction with others in virtual environment , can be improved. CVE engines need high computing resources, therefore, in past years, scientific communities chose 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 wide spread peculiarity of CVE engines if 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 into the collaborative session. In this paper we advocate 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 projected and developed a framework called WebTalk04 (WT04) [3] [4] able to generate and allow the fruition of educational collaborative sessions. This was obtained by giving the users the possibility to connect to the 3D environments through the web browser using a low resolution renderer (using the Adobe Shockwave player plug-in [5]) and using poor quality graphic performances. Thank to this framework throughout the years , hundreds of sessions in projects such as Learning@Europe (www.learningateurope.net) and Storia@Lombardia (www.storiadellalombardia.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 but they will, only differ in the specific contents of each single session still maintaining high graphic performances measured in terms of rendering speed and overall quality of generated images.

State of the art

We made a closely investigation following two specific routes in order to analyze the state of the art :

- One route related to the articles made by the scientific community (we analyzed the ACM and IEEE on-line archives);
- One route related to commercial systems.

The analysis was made comparing, for each system, the performances in terms of use of rendering technologies and its configuration degree. IRRLICHT [9] is an open source graphic engine optimized from the point of view of the rendering quality and the 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 to collaborative features or to external configuration to make collaborative experiences possible. Ultima Online [8] (often shortened UO) is a fantasy MMORG developed by Origin Systems, a software house recently bought by EA games. UO client offers the player the possibility to play both in 2D isometric graphic ang in 3D isometric with the optimized use of shaders. At the beginning the game used servers who were owners of Origin in Europe and America even though several emulators have been developed some years later. The UO server systems are hardcoded; everything is specified in the source files of the emulators (managing of the behaviours of objects, skills definitions, NPC advanced artificial intelligence) with a minor possibility of personalizing via an external configuration file. World of Warcraft [7] is an MMORG fantasy videogame which is only playable through Internet paying a fee. It is developed by the Blizzard Entarteinment and was published in 2004. The game has a 3 dimensional graphics which enables the player to go into its virtual environments. World of Warcraft is the most played MMORG in the world with over 9 million of active enrollments. The system is able to provide rendering taking advantage from the most recently graphics accelerated cards (optimizations such as "per-pixel shading" and "per-vertex shading" are used). It is impossible to externally configure the game session. Second Life [6] is probably the best known product. Developed by Linden Labs, allows the users who are connected to interact and perfectly imitate the

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

interactions which would happen between users in real world. The system uses a renderer based on OpenGL but that is hardcoded inside the client. It is possible to configure only a small number of environmental parameters through external files of configuration, but the operations are very difficult and slow. Moreover is not possible to configure interaction rules inside environments. WebTalk04 is a collaborative engine developed between 2005 and 2006 by the research group I belong to , in the GSA-Lab of the University of Salento. It is a collaborative framework able to provide a large quantity of collaborative experiences which can be configured by an XML file. In the past few years, this framework, was used in projects such as Learning@Europe, Storia@Lombardia, Learning@SocialSports providing thousands of collaborative experiences within e-learning. As said before, this framework uses a non optimized (from a performances point of view) rendering engine provided by the Adobe Shockwave player plug-in.

Taking into consideration the motivations and the state of the art study, we realized that there is a lack of high performances and highly configurable systems and therefore we have projected and developed OpenWebTalk, a new collaborative framework to fill the existing gap.

OpenWebTalk Framework

The architecture we present is the result of an optimization process we have done on our first "Openwebtalk architecture"[7]. Thanks to this new approach we optimized the behaviours management introducing a module (the behaviour manager) to simplify the event-action management during the collaborative sessions.

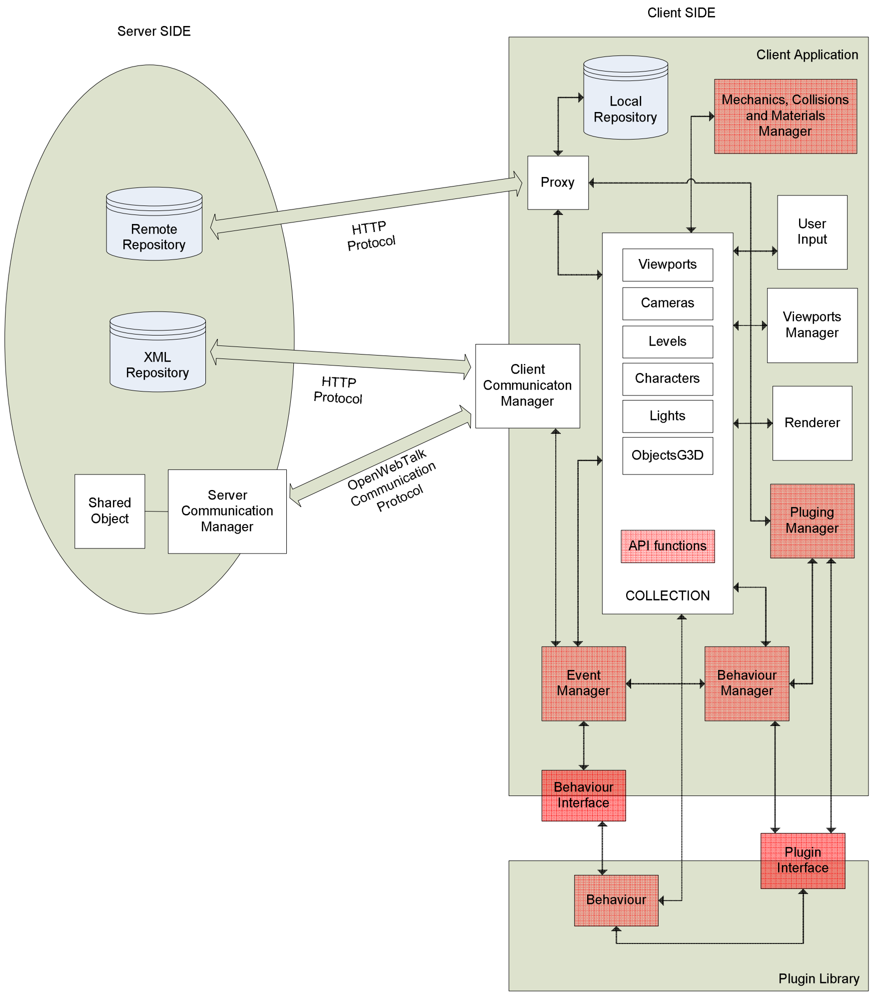
The OpenWebTalk architecture is deeply based over a client/server paradigm and exploits an application server hosting a Web Server for static and dynamic contents and a server-side Daemon (the OpenWebTalk server communication manager) for sharing data in cooperative and distributed applications.

While the server is listening to client connections and distributing events between the participants, maintaining a centralized state repository, the client side is composed by 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 event generated in the same moment from other clients connected to the same environment, are distributed by the Server Communication Manager to the Client's Communication Manager on client side, which manage them 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 each other during the on-line sessions.

All the system architecture is 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 as well as the shared environment.

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it



(Fig 1: OpenWebTalk architecture in the large)

The architecture of OpenWebTalk is shown in Figure 1. Let us now describe all its main components:

- *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 [10] 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 performances 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 performances. The "collision module" computes in real-time the collisions between the entities in the virtual world introducing physics in the virtual environment. The proxy module manages the object downloading and caching features, using a smart optimized 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 by event and behavior managers that control the collaboration features. We show in detail what is and in which way does it work in the section called "OpenWebTalk Configuration and Collaboration Model"

- *Server Side*

Contents Repository: stores all the geometries and the textures, created at design time.

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

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

Communication Server: it traces the actual shared state, keeping track, for example, of positions, rotations, and "state" in general all the avatars and objects. Any shared state change is sent to all the clients that 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.

The render module

As said before, rendering performances and optimization are very important for a real time 3d collaborative framework. We decided to implement the rendering module of the framework basing it on G3D optimized rendering module. Thanks to the G3D performances and capabilities, we can render full-screen virtual environments in real-time, using optimizations like Shaders [11], Shadow maps and shadow volumes [12] programming. Moreover we can control at run time the hardware acceleration performaces so as to optimize the frame quality. In the render module we introduced optimization that can be grouped into two different classes:

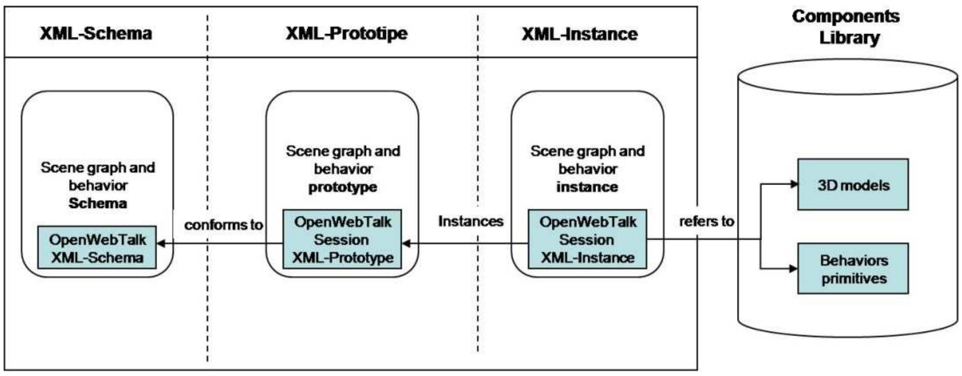
- Image quality optimizations
- Performances optimizations

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

Level of Detail and Frustum culling optimization are used during the performances optimizations to improve the rendering speed (measured in Frames Per Seconds [FPS]). Our render module (based on the G3D render module) selects the right detail level basing not only on the standard parameter as distance and visibility but also on the overall performances of the system.

OpenWebTalk configuration and collaboration model

To convey our virtual environment a high sense of virtual presence, leaving the full graphic driven paradigm to switch towards a new one that takes the overall application design into account and to provide a set of interaction features to drive the collaboration among users in the specific way the designer was intended to do. OpenWebTalk proposes a **declarative format** [14] to reduce programming need for non expert designer of the collaborative experience. 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 on how the virtual word is when it is generated, and from a dynamic point of view regarding on how it can evolve during the collaborative session through a declarative description of the Event Conditioned Actions (ECA) that can be performed in the virtual environment.



(Fig 2: OpenWebTalk declarative system levels)

In figure 2 we can see the OpenWebTalk declarative system levels composed by:

1. Scene graph and behavior 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.

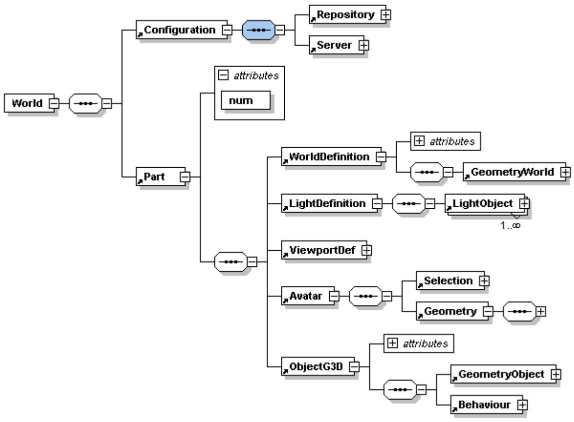
[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

2. Scene graph and behavior instance prototype: a XML document representing the skeleton of a collaborative session where designer has already fixed geometries forming the virtual environment as well as the main users' interaction rules.

3. Scene graph and behavior instance: the final and XML instance completed with all information and contents depending on the specific context (specific users involved, specific target of the experience, particular topics given).

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 that interprets the declarations and provides to instance the right components taken from a library (from one hand it instances the right 3D models in the world, on the other it sets up the collaboration's rule that will govern the shared experience).

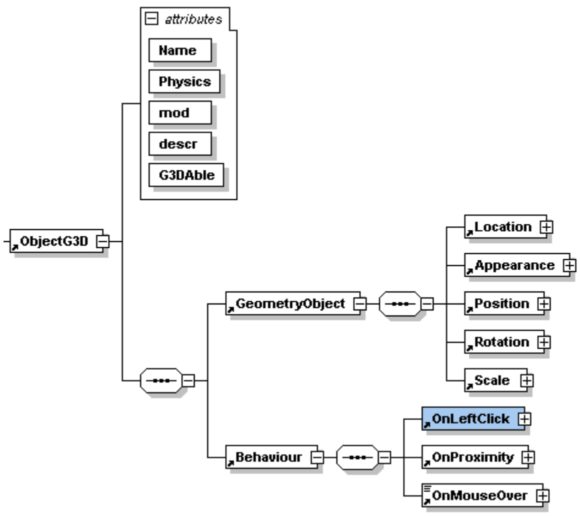
The OpenWebTalk scene graph is organized as a sequence of 3D environments called parts in which users can navigate moving from one to another simply colliding to special interactive objects, often in form of ports or gates working as teleports , thus causing the unloading of the current part rendered by the engine and the loading the next one.



(Fig 3 – Configuration XML schema)

Looking at figure 3, 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, being composed by the <avatar> element and a list of <objectG3D> elements. Each part is characterized 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 by a <geometry> element (describing geometry model, position, rotation …) and a <behavior> element (describing the possible interaction rules).



(Fig 4 – The ObjectG3D XML node)

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

In OpenWebtalk, user-to-user or user-to-object collaboration issues are intrinsically correlated to interactions' primitives that we call virtual actions supported by the environment as well as its capability to control and drive collaboration in a specific and well defined direction. In our model, interaction is achieved and controlled through the mechanism of Event Conditioned Actions.

The Plug-in Behaviour system

The plug-in behavior system we present provides the architecture with an use more optimized of the event-action paradigm. Thanks to this choice, every behavior is developed and codified in a file that will be loaded by the architecture at run time. This involves optimizations on two separate fields:

From one hand, we have an optimization from a performances point of view. The framework will load only the plug ins that are useful during the action; to the other we can add lots of behaviours simply developing new plugins without the necessity to rebuilt the entire framework. This is important also from an economical point of view.

The plugin behaviour module is dedicated to the management of the files related to the plugin sub-system.

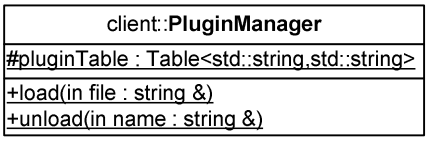
Each of the plugin used in a Part will be loaded during Xml file parsing of World Description.

Each element XML *Action* posseses a *url* attribute which allows *Client Proxy* to get its related file plugin from *Remote Repository* and to copy it in the local repository .

```
<OnProximity Distance="4">
    <Action type="LightOnOff" url="plugins/LightOnOff.dll" light="light1"/>
</OnProximity>
```

Once the file plugin is in *Local Repository*, the *Plugin Manager* module , will dynamically load the library recalling/calling the system function *LoadLibrary*, through the *load* static method

We will now describe the prototype of the class which implements such a module (shown in figure 5).



(Figura 5: The PluginManager class)

The DLL file which implements a behaviour exposes the *getInfo* method through which the name of the implemented behaviour can be obtained. The DLL file sets out/exposes also the *pluginAlloc* function which make the pointer return to an instance of the *Behaviour* class . The address of this function will be obtained using the *GetProcAddress system function* and the pair "name of the plugin" – "pathname of the plugin file " will be added to the *pluginTable* in order to receive information on/about plugin files that have already been loaded (useful to release the DLL library and to avoid multiple loadings of the same plugin). Moreover the couple "plugin name"**-** " address of the *plugin Allo* function" (useful to allocate a Behaviour by name) will be added to the *BehaviourManager::procTable)*

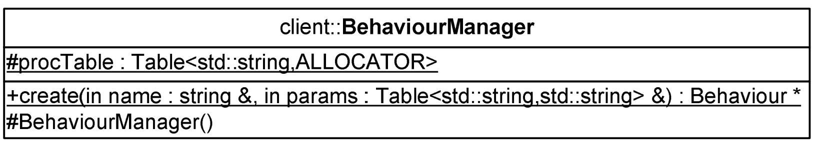
The *unload* static method will download the library from the memory and remove the items that the load method had added in both the *pluginTable* and in the *BehaviourManager::procTable.*

Behaviour Manager

This module is being implemented by the homonymous class. *Behaviour Manager* is necessary to create instances of *Behaviour* class starting from the name of the behaviour.

This is possible thanks to *procTable* which matches the names of the loaded plugin with the addresses of the related builders (note: *pluginAlloc* is actually a static function which is implemented in DLL library)

.



[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

(Figura 6:The Behaviour Manager class)

*typedef Behaviour * (__stdcall *ALLOCATOR)(GlobalCollection &, const Table<std::string, std::string> &);*

The *create* method , the only one of/in this class, is static and is deputed to work out the address of the *pluginAlloc* function related to the plugin specified by the *name* parameter and to call the function specifying as parameters the reference to the collection of global objects and the reference to the table of XML attributes obtained from Action element in World Description file.

The *params* parametre matches the name of each single attribute with its own value.

The very management of behaviours is made by a class named "Behaviour" that acts as an interface and is to be implemented in the plugin file.

Through this interface application classes can interact with behaviour classes contained in DLL calling generic behaviour methods.

The *execute* virtual method is to be implemented in DLL and is called by *raise* function of Event instance which behaviour is associated with.

The classes that implement this interface will have to define a builder. The latter will accept both a reference to the collection of shared objects and the table of attributes of XML *Action* element present in the session descriptor file

Early Results

As said before, the framework we presented in this paper is the result of an optimization process done on a previous version of our OpenWebTalk system. Thanks to this optimization our framework is now more simple, flexible and reliable. Unfortunately, we are planning a test session but we do not have still tested the engine with lots of users. We present in this paper the results of an early test session made with the help of 2 teams of students.

The FRAPS tool was used to make this test. Its Benchmark function in fact, can detect the FPS (Frames Per Second) value and save the statistics in three different modes: MinMaxAvg (benchmark duration, total frames, maximus, minimum and average frame-rates), FPS (rendered-frame value per each benchmark second), and FrameTimes (time, expressed in milliseconds, when the frame has been shown during benchmark). The tests have been made on three different types of computers: two notebooks (one with very high graphic performances and one with lower graphic performances) and one desktop (with a very common configuration).

The hardware specification of the systems we have used during benchmarks are detailed below:

	Machine 1	Machine 2	Machine 3
Name and model	Fujitsu-Siemens AMILO D 1840W	Fujitsu-Siemens AMILO Pi 1536	Dell Dimension 9150
Type	Laptop	Laptop	Desktop
Processor	Intel® Pentium® 4 HT	Genuine Intel® CPU Core Duo T2400	Intel® Pentium® D 930
CPU frequencies	3.00 GHz	1.83 GHz	3.00 GHz
RAM	512 Mb	2048 Mb	2048 Mb
Video Card	ATI mobility RADEON 9600 a 128 Mb	ATI mobility RADEON X1400 a 512 Mb	NVIDIA GeForce 7300 LE a 512 Mb HyperMemory
Video Driver	ATI technologies Inc	ATI technologies Inc	NVIDIA
Driver Version	7.91.4-030724a-010417C	8.391-070626a1-049711C-ATI	5.72.22.41.31

(Table 1. Machines hardware specification)

All the benchmarks have been performed on a virtual environment according to the following configuration:

Number of object: 70;
Number of lights: 5
Two cameras whose configuration was set on default values
4 avatars connected
A view made up of 552.200 vertexes, 5.404 triangles and 346 primitives

The test has been made changing the setting in such a way as to make the rendering in three different resolutions (640x480, 800x600, 1024x768).

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

	Risolution (pixels)	Frames	Time (ms)	Mi n FPS	Max FPS	Avg FPS
Machine 1	640x480	1779	60000	24	35	29.500
	800x600	1789	60000	29	31	29.817
	1024x678	1186	60000	19	23	21.000
Machine 2	640x480	5664	60000	90	96	94.400
	800x600	3804	60000	65	69	67.000
	1024x678	2636	60000	44	46	45.000
Machine 3	640x480	1801	60000	33	35	34.000
	800x600	1801	60000	31	33	32.000
	1024x678	1201	60000	20	22	21.000

(Table 2. Benchmark results)

The tests we have presented have shown that the OpenWebTalk framework is able to maintain high performances with different resolutions and on different systems. It is particularly important to point out that referring to the three different resolutions the average value showing the FPS never goes under 21 FPS (reaching the minimum value of 21.000 with a resolution of 1024x768 pixels on the computer with the worst hardware characteristics). To measure the time necessary for the creation of an XML instance related to a typical session, we asked a user skilled in the configuration of collaborative sessions using our framework , to configure a session made up of four parts with 30 different objects and two different behaviors for each object starting from an XML skeleton. The test has been repeated 10 times. The average time necessary for the creation of XML was 12.5 minutes.
An additional set of tests has been made comparing the performances in terms of FPS of the developed framework with those of the original OpenWebTalk framework.
The results we obtained are shown in the table below:

	Resolution	Frames	Time (ms)	Min	Max	Avg
OpenWebtalk*	640x480	1678	60000	120	237	178
	800x600	1690	60000	103	198	150
	1024x678	1086	60000	93	128	110
OpenWebtalk	640x480	1779	60000	113	225	169
	800x600	1789	60000	93	175	134
	1024x678	1186	60000	84	112	110

(Table 3. OpenWebTalk* vs. OpenWebTalk compared test results)

The test has demonstrated not only that OpenWebTalk* performances in terms of FPS are superior if compared to the OpenWebTalk performances, but it also has shown that the new version of the framework “suffers” the increasing of the resolution less than the original. This is possible thanks to the optimization we have made in terms of loading/unloading of the behaviours during the experience sessions.

Future Works

The OpenWebtalk Architecture has a high rendering graphic performance also on computers with poor hardware characteristics. Another interesting result was obtained in reference to the time needed by an experienced architect to make an instance of collaborative session starting from a skeleton. All this result has been obtained simplifying the architecture and optimizing the behavior management.
In the future, we are going to take the following steps in three different routes:

- Improvement of rendering quality: we will support new effects and new optimizations during the rendering
- Increase of the features linked to collaborativeness: we will insert new behaviours and new modes of interaction
- Integration of the frameworks with Virtual Reality (VR) devices: some modules will be inserted in the architecture to enable it to be interconnected with some devices, such as gloves and head-mounted-displays, stereoscopic rendering) etc. , in order to increase the sensation of going into environments

Acknowledgements

The authors would like to thank the Prof. Ing. Domenico Laforgia, rector of the University of Salento, for his permanent strong support; the entire staff of Hoc (Hypermedia Open Center) laboratory, Polytechnic of Milan, and its Scientific Director Prof. Paolo Paolini. We would like to thank also all the GSA-Lab (Graphics and Software Architecture Lab.) staff, and all the students of the Computer Graphics and Software Engineering courses.

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it

REFERENCES

[1] http://gsalab.unile.it

[2] Barchetti, Bucciero, Mainetti, Santo Sabato: "WebTalk04: a Declarative Approach to Generate 3D Collaborative Environments" – in proceedings of short papers of 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, 3rd Eurographics Workshop on Graphics and Cultural Heritage, p 55-60, Pisa, Italy, November 2005.

[3] Barchetti, Barbieri, Bucciero, Mainetti, Santo Sabato: "WebTalk04: A Framework To Support 3D Collaborative E-Learning" in proceedings of the The Fifth IASTED International Conference on Web-Based Education, p. 13-18, Puerto Vallarta, Mexico,

[4] Brooks Jr., F. P. (1999). "What's Real About Virtual Reality?", IEEE Computer Graphics And Applications, 19(6), 16

[5] Burdea, G. and P. Coffet (2003). Virtual Reality Technology, Second Edition. Wiley-IEEE Press.

[6] Barchetti U., Bucciero A., Mainetti L., Santo Sabato S. Webtalk04: A declarative approach to generate 3d collaborative environments. In Short Presentations of The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST. Eurographics, 2003.

[7] Barchetti, Bucciero, Mainetti, Santo Sabato - OpenWebTalk: A FRAMEWORK TO SUPPORT HIGH PERFORMANCES AND HIGH RESOLUTION 3D COLLABORATIVE EXPERIENCE - INTED 2008 Conference - Valencia, Spain - March 2008. ISBN: 978-84-612-0192-1, Pag 112

[8] Alberto Bucciero, Luca Mainetti, Thymoty Barbieri, Ugo Barchetti and Stefano Santo Sabato. Webtalk04: A framework to support 3d collaborative elearning.In Proceedings of Web-based Education, page 26. Acta press, 2006.

[9] Adobe Macromedia. HTTP://WWW.ADOBE.COM, 2006.

[10] www.secondlife.com

[11] www.worldofwarcraft.com

[12] www.ultimaonline.com

[13] irrlicht.sourceforge.net

[14] http://g3d-cpp.sourceforge.net/

[15] Holger, Regenbrecht - Haller, Michael - Hauber, Joerg - Billinghurst, Mark: interfacing remote collaborative virtual environments with table-top interaction, CSSR2006 Austria University of Applied Sciences, Frank-Fritsch-Straße, Upper Austria, Austria, November 2006

[16] PETTIFER S., MARSH J. : A collaborative access model for shared virtual environments. In Proceedings of IEEE WETICE 01, pages 257-272. IEEE Computer Society, June 2001

[17] Haiyang Yua, Farouk Yalaouia, Ėric Châteleta and Chengbin Chua, Optimal design of a maintainable cold-standby system. Elsevier 2007, Volume 22

[18] Barchetti U., Bucciero A., Mainetti L., Santo Sabato S.: WebTalk04: a Declarative Approach to Generate 3D Collaborative Environments. Short Paper of VAST 2005, Pisa, Italy, November 2005.

[19] A 3D Virtual Environment for Social Telepresence Steve DiPaola VINTED 2007 - Technical University of British Columbia, CANADA, pp 153-159

[20] Jeff Lander, optimizind a 3D world with dynamic shades. Game Developer Jurnal, November 2006, pp.3-12.

[21] www.learningateurope.net

[22] www.storiadellalombardia.it

[23] Ulf Assarsson and Tomas Möller. Optimized View Frustum Culling Algorithms for Bounding Boxes. Jurnal of graphics tools, 2006, Volume 5, Number 1.

[24] Max Wagner, Frustum Culling optimization,

[1] E-mail: [ugo.barchetti, alberto.bucciero, antonio.capodieci, luciano.capone, luca.mainetti, stefano.santosabato]@unile.it