Creating Multiuser Web3D Applications Embedded in Web Pages

Xandre Chourio, Francisco Luengo and Gerardo Pirela

Scientific Modeling Center (CMC), University of Zulia
Maracaibo, Zulia 4004, Venezuela

Abstract
It is not common to find web pages that show interactive multi-user 3D virtual environments as part of their contents without requiring special plug-ins for the web browser to be able to execute such an application. This paper presents a new architecture based on Java technology to create web portals which include 3D virtual scenarios that many users may share whose actions in that environment can incur in automatic updating of the rest of the portal's contents, without requiring any special additional software to be displayed by or installed unto any browser.
Keywords: *Web3D, java, web portal, client-server architecture.*

1. Introduction

Despite the increase of 3D applications on internet, the web is still being substantially a 2D world. It is uncommon to find web pages which include a 3D interactive environment among its elements. The usual mode of displaying information is still text, image, or video rendering on the webpage, and the predominant browsing mode remains via hyperlinks associated to those elements. Despite the great advancements in current technology, there still exists a considerable gap between the Web as we know it today and the Web3D as we imagine it.
Web applications use web browser as user interfaces. Their way of displaying information has evolved significantly from their beginnings. As Internet's presence and power, as well as diversity of contents, increase so does the user community's demand for more intuitive web applications with regards to information retrieval and handling. In fact, 3D web environments have existed for decades and languages such as VRML have tried to become the equivalent of HTML for 3D interactive environment visualization on the web.
A technology does not currently exist that offers the kind of information interaction and rendering capability found in desktop applications. The main reason for this is the heterogeneous network environment on which Internet operates, as well as the security mechanisms needed to execute applications safely.
Current research and development trends allow us to visualize a new generation of web browsers with powerful GUIs which are able to handle and render realistic, interactive 3D content. But, in the meantime, many other research efforts point towards determining how to choose and combine current technology to achieve more effective and efficient development of web applications able to implement such interfaces as such technology combinations díctate the needs, uses, and capabilities of future web browsers.
In this article we propose a platform based on Java technology for publishing virtual 3D interactive, multiuser environments capable of interacting also with the rest of the web page's content and which can be viewed on standard web browsers without having to install any plugins. We additionally present a web portal to show the platform's functionality.
In section II we comment on Web3D and the attributes any Web3D application must possess. Section III presents an evaluation and selection of tools. Sections IV and V describe the platform we propose. Finally, conclusions are shown in section VI.

2. Web3D

Web3D groups any programming language, protocol, file format or any other technology that allows to integrate interactive 3D virtual environments as part of web page contents [19].
3D graphic applications for the web formally started with the virtual reality modeling language (VRML) which appeared in 1994 and has become since a standard for creating, transmitting, and representing 3D objects and 3D environments on the web. Later, in 2001 the Web3D Consortium [20] announced the release of X3D as the new standard succeeding VRML. X3D offered complete back-compatibility with its predecessor and included significant

improvement such as NURBS extension, humanoid animation, morphing, and the use of XML-like syntax, making it extensible and facilitating its use with other applications.

However, a web portal's vistosity is based not only on offering the user attractive elements, but also on how such elements behave under different events [1]. This functionality is commonly achieved by using programming languages, and hence companies such as Sun Microsystem Inc. have developed APIs for 3D graphics like Java3D [7], which is not oriented for web use but which can be combined with Java-Applets as an alternative for developing 3D web applications. Similarly, other Java APIs have emereged which try to take advantage of graphic hardware through OpenGL, such are the cases of JOGL [5] and LWJGL [9] among other products which do not require installing any additional software component, like jGL [2].

Java is one of the strongest languages for web application development due to its versatility, robustness, and multi-platform support. Its usability in the case of Web3D can be appreciated in applications such as Cortona Jet [3] and Shout3D [17] which combine proprietary technology with java-based interfaces to provide any web browser with capability for 3D interactive graphic rendering without requiring any special software installation. Another prominent example is the Project Wonderland [15] which was entirely developed in Java and uses Java3D and other technologies [8,14] to provide a set of tools which allow creation of virtual collaborative worlds in which the users can communicate and share applications in real time.

Currently, big companies and consortiums are renewing their interest in this technology, promoting the development of new tools for Web3D. In this particular, Google has recently launched O3D, an API written in JavaScript which allows construction of virtual, interactive 3D environment for the web [11]; however, this API seems to be oriented towards developing simple (low complexity) environments and this may become a limitation in the future. On the other hand, the Kronnos Group is currently developing WebGL, an API similar to O3D but with native OpenGL support [21]. Another example of such new 3D graphics technology for the web is Ajax3D [13], which also uses JavaScript for drawing and interaction.

Even though the term Web3D refers to technology that allows to take 3D virtual environment to the Web, recently the term has also been used to encompass applications developed from such a technology, that is applications that deliver 3D content into Web [10,18].

A 3D application for the web exhibits specific properties according to its nature and which determine its functionality. Such properties and characteristics are listed below.

Such properties and characteristics are listed below.

a. Hardware Independence – Due to the heterogeneous environment that surrounds Internet, it becomes necessary that applications exhibited therein may run into any platform.
b. No software installation required – To protect users and ensure their safety, no installation should be required on the side of the client to run such applications.
c. Graphic capability to create 3D virtual environments – Tools, such as OpenGL, add realism to the environment with the use of techniques such as shadowing, texturing, NURBS, etc., in order to create a web experience similar to that which desktop applications offer.
d. Multiuser – Enabling sharing a common environment and allowing communication and interaction among several users.
e. Highly interactive – The application should be versatile and offer the user easy control via a rich, friendly interface.
f. Coupled with Web page contents – This implies that the application not only be embedded in a web page but that it also be able to complement itself interacting with other contents and resources in the same webpage, dynamically.

General web applications show many of the characteristics described above, except for graphic capability to create 3D virtual environments (point c in the list), which is specific to Web3D applications. However, access to this characteristic is precisely what drives developers to sacrifice some other attributes in the list. For example, in order to take advantage of the client's graphic hardware power, many Web3D applications require for its visualization that additional software be installed or that special applications be used. On the other hand, these applications tend to be independent from the rest of the hosting webpage's content; that is, the interaction in it does not affect the rest of the elements on the page (point f in the list) and hence, the application is not truly integrated to the web portal or it simply runs on a separate window.

We describe in the following sections implementation details of a Web3D application which show all the attributes listed above.

3. Technology Selection

Many researchers opt to combine several different tools to develop their Web3D applications because no single tool provides all the functionality they expect. Instances of this are products which integrate XHTML, scripting languages (JavaScript, Python), 3D markup languages (VRML, X3D), web services, and AJAX-based technology to develop multiuser environments [4,12,23].

On the other hand, the Java platform offers developers the possibility to create embedded applications in web pages with high-level programming language and support for multiplatform development, security, communication, database connection, interaction, access to DOM (Document Object Model), among other capabilities [6]. Additionally, the Web3D Consortium promotes a library for X3D support written entirely in Java [22].

Both approaches allow the development of applications which satisfy the characteristic described in the previous section. However, since Java is a compiled programming language, its running time is faster than that of scripting languages such as JavaScript; moreover, Java development environments make it easier to create versatile, wide-range applications, which hence makes Java the ideal tool to develop Web3D applications as defined in section II.

However, Java's greatest drawback is that it does not offer native support for 3D graphic generation. One way of overcoming this drawback is using jGL [2] – a library for 3D drawing for Java which emulates OpenGL.

To illustrate jGL's scope for drawing 3D environments, we show below a performance comparative analysis against JOGL [5], a Java API which implements OpenGL and takes advantage of the client's graphic hardware power (for which it needs to be installed unto the computer where the application will run). Figure 1 shows the time the API takes to process a polygon grid at different resolution. Figure 2 shows the effect illumination and texturing techniques have upon both API's performance.

As expected, when taking advantage of the client's graphic hardware power, JOGL shows significantly superior performance as compared to jGL. However, these results allow us to set a polygon bound on 3D modeling and representation to achieve acceptable running time with jGL.

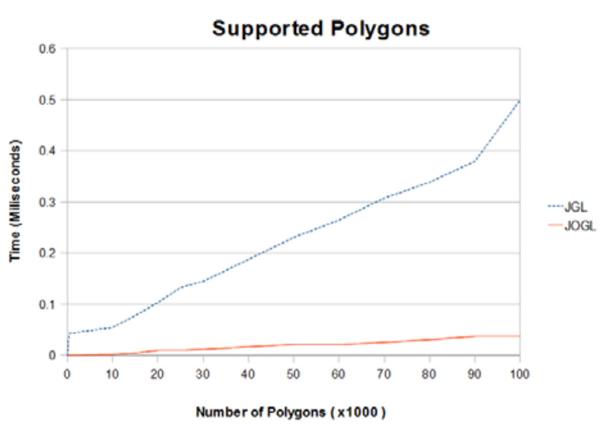

Fig. 1 jGL and JOGL's performances

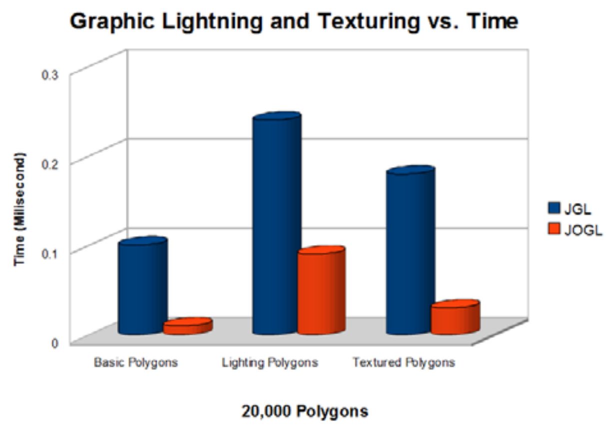

Fig. 2 Drawing, lighting and texturing poligons with jGL and JOGL.

4. Architecture

We describe below implementation details of a Web3D portal which meets all the requirements mentioned in Section II and which we call henceforth SMW3D (Multiuser Web3D System)

SMW3D relies upon three fundamental pillars, rendering, communication, and information. These pillars in turn orbit around the Java Virtual Machine – a robust, secure platform which current support allows it to be used in almost any operating system and web browser.

The structure of the portal consists of three java applets and other DHTML elements embedded in a web page (see Figure 3A). The main applet (Figure 3B) renders the virtual 3D environment, letting the user navigate through this environment, visualize other users who may be connected to the portal, and interact with them. This applet can also communicate with the rest of the web page elements to respond to the interaction being carried out. Two such elements are the other two applets (Figure 3C); one of them shows a 2D map of the environment with every connected user's location in it, and the other allows users' communication via chat. Figure 3D shows the rest of the elements of SMW3D – a frame to show a webpage and multimedia contents.

Additionally, SMW3D is built under a client-server architecture which deploys four server applications that provide data to make the portal fully functional (see Figure 4). These server applications are a Web Server (which attends to web page requests), a User Server (which keeps track of users connected to the portal), a Data Server (which provides data for 3D environment rendering and avatars), and a Communications Server (which allows communication among users). All those servers (except Web Server) were made using the Java Platform's main features (see [6]).

The SMW3D's architecture and comunication schema among their componets are discribed next.

4.1 Web Server

It is a standard web server (Apache, IIS, or any other) which hosts the portal and which answers the web browser's requests (when the portal is needed) as well as the main applet (to update the portal's DHTML elements with regards to the user interaction with the 3D environment).

4.2 User Server

This server is in charge of keeping track of all connected users, their position and orientation in the virtual environment, as well as their IP address. The server recieves this information from every connected client's main applet every time a user's position within the environment is updated.
Every time the server receives an update of a client's position, orientation, or a new client's connection, it retransmits it to the rest of the connected clients. The application then is able to correctly draw all the avatars within all open portals' scenery. Additionally, the server sends the updated list of connected users to all other servers for them to correctly run their respective functions.
Nevertheless, establishing peer-to-peer communication may overload the server due to constant traffic when many users are connected and in continual motion. An alternative could be a mixed communication scheme that includes multicast technology to avoid server overload; however, for security reasons, applets do not support multicast and so we opted for a broadcast communication scheme which also allows for more manageable data package size.

4.3 Data Server

This server is in charge of providing all open portals main applets the graphic information they need for the correct rendering of the 3D virtual environment (vertices, polygons, colors, lighting, textures, area identification, information associated with each area, delimitation of each area in the environment, etc.) In order to reduce the amount of data to be delivered for graphic rendering as users move through the environment, a database was created using MySQL DBMS and implementing a quadtree structure [16]. This data management technique is commonly used in videogames.

4.4 Communications Server

This server allows communication among users via chat, taking advantage of the connected user list provided by the user server. It manages connection better by listening for open portals' chat applet requests through a different port than that used by the User server. It is important to mention that even though the User Server keeps and provides the list of connected users, the Communications Server also keeps its own list of chatting users (or users available for chat) and it is this latter list the one displayed in the chat applet as every user is autonomous to use this module or not without this decision affecting the user's status in the main environment

5. Multiuser Web3D System

Figure 4 illustrates this Multiuser Web3D System (SMW3D) which works as follows. The Web Server delivers to each connected user's portal all the resources associated with it. Once the portal has been loaded up unto the client's web browser, the main applet communicates with the User Server requesting use of the communication channel via an "alias" for the user (who must input such an alias along with gender and chat usage status). The User Server also checks the alias to avoid duplication. Once connection has been established, the main applet communicates with the Data Server requesting information about area in the virtual environment where the user is located so it can be rendered on the scene, along with information to be published in other components of the portal. Simultaneously, the User Server shares the alias with the Communication Server to establish a communication channel for the user with this resource at the user's disposal. At the same time, the applet which shows the 2D map (Figure 3C) is constantly listening to the User Server channel to update all users' positions in its map, differentiating its own user from the rest. The main applet, in turn, is also listening to the User Server for other users' position/orientation updates in the environment. When such updates occur, it compares the new information against its current information to render the corresponding avatars updates, generating corresponding animations (walking, turning).
Communication between the main applet and the User Server is carried out every time the user moves through the virtual environment. Similarly, when the user accesses an area for which the main applet has no graphic information to render, it talks to the Communication Server requesting information for the new area to be accessed.
Furthermore, when the user accesses certain areas of the virtual environment, said areas may have associated, informative content, maybe even a URL, which the main applet distributes to the other elements of the portal in charge of publishing it. For example, when the user accesses a certain office within the virtual environment which has a web page associated to it, the main applet establishes a communication with the HTML element

destined for that exact purpose via a DOM. There are many ways to communicate with a DOM in Java (JsObject, DomAPI, and applet-specific functions). In our case, we use applet-specific functionality, as shown below.

```
try {
   applet-maestro.getAppletContext().
   showDocument(newURL("javascript:
      carga-dinamica(\""+url+"\")"));
}
catch (MalformedURLException me) { }
}
```

Access to DOM allows the main applet to update any portal element according to user interaction within the virtual environment.

6. Conclusions

The main objective of this research was developing multiuser Web3D portals – web pages with embedded applications which render 3D virtual, secure, multi-user, interactive environments. The authors reviewed different Web3D technology alternatives and other contributions to the fields which ended up justifying the use of the Java platform as a viable, ideal technology to build Web3D portals according to the criteria presented in Section II.

As a result, we created a web portal for a University dependency which shows several elements related to such dependency such as the start page, a 2D map of its facilities, and a 3D environment (among other information). The application assigns an avatar to each connected user for his or her representation in the 3D virtual environment and identification therein. Users may navigate through the virtual environment which will automatically update all the other elements of the portal (contextual web page frame, banners, etc.) relative to user's location. Moreover, users which are connected to the portal can also communicate among one another via chat.

Even though many developers combine different technologies to take advantage of each one of them, the Java platform offers a wide variety of tools to create complex applications which can be used to build colorful, dynamic, multi-user 3D virtual environments, emphasizing speed at runtime (or application response time), interactiveness, interconnectivity, and security. Furthermore, it is faster than those scripted languages. However, lack of native support for 3D rendering is still Java's only drawback.

The current tendency points towards web browsers with powerful GUIs which are able to take advantage of graphic hardware capabilities, where web contents may be interpreted, and which may deploy realistic, interactive 3D content without having to download any complementary applications of dubious provenance.

References

[1] BOUSSINOT F., SUSINI J-F., DANG TRAN F., and HAZARD L. A reactive behavior framework for dynamic virtual worlds. Proceedings of the sixth international conference on 3D Web technology. pp 69-75. 2001.

[2] CHEN B.Y. and NISHITA T. jGL and its Applications as a Web3D Platform. In ProceedingsWeb3D 2001. pp 85-91. 2001.

[3] Cortona Jet, Parallel Graphics, Inc. [online] on <http://www.parallelgraphics.com/products/jet>

[4] FRANCIS B., and STONE R. ebScylla: a 3D web application to visualise the colonisation of an artificial reef. Proceedings of the 14th International Conference on 3D Web Technology. pp 167-175. 2009.

[5] Java Binding for OpenGL [online] on <http://kenai.com/projects/jogl/pages/Home>

[6] Java SE Technologies at a Glance [online] on <http://java.sun.com/javase/technologies/index.jsp>

[7] Java3D [online] on <https://java3d.dev.java.net/>

[8] jME (jMonkeyEngine) [online] on <http://www.jmonkeyengine.com/>

[9] Lightweight [online] on <http://lwjgl.org/>

[10] LUENGO F., CONTRERAS M., LEAL A., and IGLESIAS A. Interactive 3D Applications Embedded in Web Pages. 4th Internacional ConferenceComputer Graphics, Imaging and Visualization. pp 14-17. 2007.

[11] O3D API [online] on <http://code.google.com/intl/in-IN/apis/o3d/>

[12] OSTROWSKI D. A Web-Based Interactive 3D Visualization Architecture. IEEE Visualization-Poster, Baltimore, MD. (USA). November 2006.

[13] PARISI T. Ajax3D: The Open Platform for Rich 3D Web Applications. Media Machines, Inc. Whitepaper. 2006.

[14] Project Darkstar [online] on <http://projectdarkstar.com/>

[15] Project Wonderland [online] on <https://lg3d-wonderland.dev.java.net/>

[16] SAMET H. Foundations of Multidimensional and Metric Data Structures. Morgan-Kaufmann, San Francisco. 2006

[17] Shout3D v2.8 Shout Interactive, Inc. [online] on <http://www.shout3d.com>

[18] STEFANO TORNINCASA. Web3D Technology applications for distance training and learning: the Leonardo project "WEBD". XII ADM International Conference. 2001.

[19] WALSH A.E. and BOURGES-SEVENIER M.Core Web3D. Prentice Hall PTR. 2001.

[20] Web3D Consortium [online] on <http://www.web3d.org/>

[21] WebGL – OpenGL ES 2.0 for the Web [online] on <http://www.khronos.org/webgl/>

[22] Xj3D - Java based X3D Toolkit and X3D Browser [online] on <http://www.web3d.org/x3d/xj3d/>

[23] YIN S. A Web-Based 3D Gaming Style Multi-user Simulation Architecture. 1st International Workshop on WEB3D GAMES 2007. pp 19-22. 2007.

Xandre Chourio works as Research Assistant assigned to University of Zulia's Scientific Modeling Center (Venezuela). He

holds a B.Sc. degree in Computer Science at the University of Zulia (2010). His fields of interest are client-server applications, distributed computing and grid computing.

Francisco Luengo is Titular Professor at the Department of Computer Science of the University of Zulia (Venezuela). He holds a B.Sc. degree in Computer Science at the University of Zulia (1992) and a M.Sc. in Computer Science at the Central University of Venezuela (1996) and a Ph.D. in Applied Mathematics and Computational Sciences at the University of Cantabria (Spain, 2005). His fields of interest are computer graphics and animation, parallel computing, artificial intelligence and numerical analysis.

Gerardo Pirela is Associate Professor at the Department of Computer Science of the University of Zulia (Venezuela). He holds a B.Sc. degree in Computer Science at the University of Zulia (1997) and a M.Sc. in Science and Computer Engineering at Oregon Graduate Institute of Science & Technology (USA, 1999). His fields of interest are Artificial Intelligence and Bioinformatic.

Fig. 3 Multiuser Web3D page.

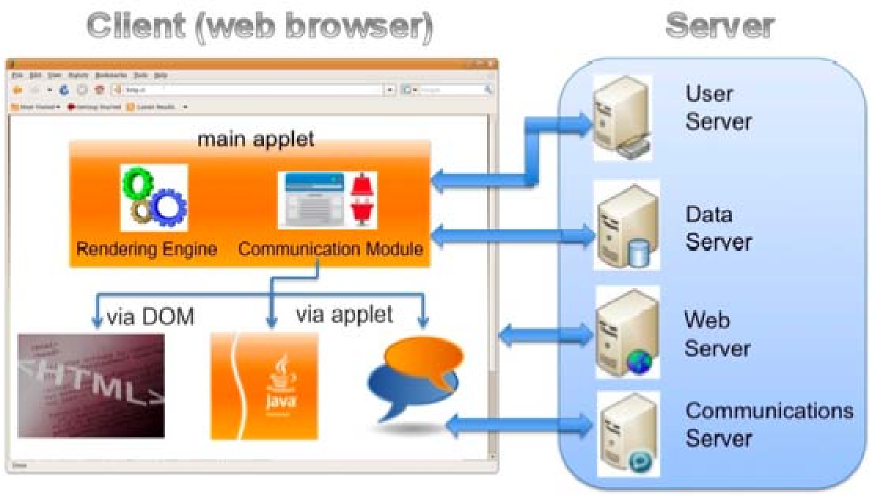

Fig. 4 Multiuser Web3D System Architecture