Dynamic, ecological, accessible and 3D Virtual Worlds-based Libraries using OpenSim and Sloodle along with mobile location and NFC for checking in

Rubén González Crespo, Sergio Ríos Aguilar,
Pontifical University of Salamanca in Madrid, Spain
Roberto Ferro Escobar, Nicolás Torres
Distrital University Francisco José de Caldas, Bogotá, Colombia

Abstract —**This paper proposes the implementation of a 3D virtual library, using open platforms such as OpenSimulator and Sloodle, applied to the integration of virtual learning environments. It also proposes their application to the creation of open libraries to share and disseminate the new dynamic nature of knowledge, in the understanding that 3D virtual worlds may contribute to the future of libraries as part of green initiatives to achieve an ecologic and sustainable planet.**

Keywords — **3D Virtual Library, OpenSim, Sloodle, NFC, GPS check-in, Grid.**

I. INTRODUCTION

TODAY, one of the most relevant topics in computer-based content management research efforts is the way of presenting and visualizing content, stepping ahead from simple static web delivery of documents, books and publications via HTTP on the Internet. Nowadays, users demand better ways of accessing new sources of information, and an evidence of this is the current evolution of Digital Libraries, making huge efforts in order to make large collections of documents available to general public.

Anyway, current prevailing paradigms show that changes are needed in the way these contents are presented to the information consumers, who are looking to interact with Web contents in a more friendly manner using richer interfaces. And provided that recent evolution in office and domestic broadband access to Internet has improved the performance of networks interconnections and hence the globally perceived speed of Internet traffic, this change begins to come true thanks to virtual worlds simulators like Open Simulator [1].

This simulator was created under the guidelines of Secondlife, released in 2007 by Linden Labs [2], and placed under the GPL license [3]. OpenSim meets the general client-server model and has a modular structure written in C# and supports multiple viewers, such as the HIPPO viewer [4] , RealXtend viewer [5] and the Secondlife official viewer itself

The dynamic nature of Web 3.0 [6] combines artificial intelligence techniques, geospatial mobility, mashups, interactivity, multimedia and 3D transformations to build a semantic database able to learn. And everything is accomplished in 3D virtual worlds, where the user can be embodied in avatars that allow travelling to and wandering many worlds constantly arising, built using freely available design tools. Secondlife was initially used as a game in a virtual world built on a commercial server, and after a few years, due to the many worlds that can be created, Linden Labs decided to release the code to allow for an open construction of virtual worlds and private islands by programmers.

One of the most promising research areas related to 3d virtual worlds technology is the study of the integration of these 3d virtual worlds with current virtual learning environments and content management systems, given the powerful emergent properties arising from such a combination.

Another prevalent trend is the use of mobile broadband in order to access any kind of information using a personal mobile device, and one promising technology now close to reaching the necessary critical-mass is Near Field Communications (NFC). This technology provides a proximity-based interface to a wide array of services, including mobile check-ins, in special check-ins to virtual worlds using smart tags deployed at strategic places in the real world.

Also, when NFC is not available, and we need to promote (or restrict) the access to the virtual world Library from within certain known premises, it is always possible to get the real-world location of the user proving his/her presence via location based services (ie GPS, A-GPS…etc), used as an alternative check-in method. Current Android and iOS mobile devices (>90% market share) provide us with location services with enough accuracy (meeting E-911 and E-112 strict requirements with a wide margin)

This paper proposes the fundamental basis for the development of a 3d virtual world based Library using OpenSim, and facing the integration of a virtual learning environment (VLE, LMS) such as Moodle, with the help of the Sloodle platform.

Fig. 1. View of an open virtual world simulator with geospatial database connectivity and information sharing in real time.

II. TOOLS FOR CREATING METAVERSES

A. OpenSim 3D modeller

OpenSim is a 3D application server that uses the same standards found in SecondLife to communicate with their users. This modeller emerges from the analysis of the structure of SecondLife viewer's source code, released early in 2007 by Linden Labs under GPL. It was reverse engineered in order to create a 3D application server, and is characterized by being free software, having a modular structure written in C # and its support for multiple viewers; to date, OpenSim is in version 0.7.4

Among the most promising features of OpenSim is the ease with which universities can customize their LMS (Learning Management Systems) [8] integrating the legacy systems with OpenSim databases and creating content that suit the needs and methodologies of each institution.

OpenSim is being currently used to design, create and easily implement 3D virtual worlds by a huge base of programmers around the world. With this tool, it is possible to create content targeted to different areas, such as e-learning, virtual tours, simulations, among others. Moreover, being a free software tool leads itself to low-cost development and allows developers to contribute to the development of new features.

B. Moodle for Web courses

Moodle (Modular Object Oriented Dynamic Learning Environment) is an open source software package for creating courses and web sites based in PHP and designed to work with different database engines, especially MySQL [9]. Moodle aims to help educators to create and easily manage online learning communities, and is designed to foster the construction of knowledge by students.

Its first version was published in August 2002 and to date the current version is 2.3. Moodle currently has over 6.856. 000 courses offered through 72.164 registered sites in 223 countries, with Spain holding the second place in World

Ranking of registered courses, and Colombia in the eighth position.

C. Sloodle project

The SLOODLE project (Simulation Linked Object Oriented Dynamic Learning Environment) [10] is a successful code that integrates multi-user virtual environments like Second Life and OpenSim with learning management platform MOODLE. Sloodle provides a wide range of tools (some still in development) that enable virtual learning worlds and immersive education, and are fully integrated with a web-based learning management system, used by thousands of educators and students worldwide.

D. Ubuntu operating system

Ubuntu is an operating system that uses a Linux kernel and based on Debian [10], is maintained by the British firm Canonical and the developer community. This distribution is intended to improve the end user experience, making it easier to interact with the operating system through an improved GUI. It is one of the most popular distributions today, with approximately a 50% share among Linux distributions.

For the development of this project, it should be noted that Linux is widely known as an operating system designed from the beginning as multithreaded as multiuser, so it does manage very safely the permissions for each user on processes and files, preventing users from accessing each other's information. Ubuntu has default user "root" disabled, so that the system repeatedly asks the user's password to install components and forces him to be especially careful with permissions to modify files and folders.

III. INSTALLATION OF METAVERSE IN OPENSIM

Below is a brief summary of the steps needed in the Ubuntu distribution in order to assemble a metaverse in OpenSim, and linking it to the Moodle platform.

A. Installing XAMPP Platform

Given that we need a web server to distribute the contents held by Moodle and that it is advisable to configure MySQL as the default database engine for a OpenSim server, we will use the XAMPP [11] Apache distribution, that includes the installation of MySQL. The XAMPP installation files for linux are available on the official website "Apache Friends". Once downloaded the file, you must deploy it uncompressed in the "opt" directory. Finally you must start the server using the command "/opt/lampp/lampp start".

One of the most powerful utilities found in XAMPP is the php-based administration tool for MySQL databases "phpMyAdmin", which can be accessed typing in a browser "http://localhost".

By default, the admin page does not have access control enabled. So, anyone with access to the server is able to change these valuable data. We recommend setting passwords. The XAMPP documentation describes how to do this.

B. Installing the Metaverse world with OpenSim

On the OpenSim download page we can find the necessary files for the installation of the server. It is advisable to mount the server on a 32-bit Linux version, since 64-bit versions often have compatibility issues with OpenSim and its dependencies.

OpenSim is designed to run on Microsoft's .NET platform. As the code and scripts used in the metaverse are based on C#, we should install the Mono platform on Linux, which has the libraries necessary to compile such languages. The required files for installing Mono on Linux are available at their official download page or they can be downloaded directly using the terminal installation commands (eg Ubuntu: sudo apt-get install mono-complete). After having installed mono opensim unzipped the file, we must open a Terminal. To start the server, just type "mono OpenSim.exe" command, with root permissions.

If there are no compatibility issues, the server should start gracefully, loading one by one all the necessary dependencies to the metaverse. If this is the first time you run OpenSim on the computer, it does not have any metaverse to begin with, so the simulator requests the creation of one, asking for the following information:

- New region name[ ]: alphanumeric character. Enter the desired name for the region.
- RegionUUID [default_UUID]: UUID. Unique identifier for this region. OpenSim proposes a default UID, so you should just press enter unless you prefer a specific UUID.
- Region Location [1000.1000]: x, y coordinates. It is the point in a coordinate plane you want for the region. Mode must be specified for non-grid overlay on existing regions region. Otherwise you can accept the default one point by pressing the enter key.
- InternalAddress [0.0.0.0]: Wildcard Mask (IP address). This address is actually a wildcard mask that allows you to restrict the range of addresses that can access the server. In most cases, should be left with default value, so that all hosts are allowed to access our server.
- InternalPort [9000]: # port (0-65535). Internal server port to be used for server access.
- AllowAlternatePorts: True or False. This feature is still in experimental stage. Default is false.
- ExternalHostname: ip address. This is the address used by clients to access the server, which should take into account some other considerations that are discussed later. To start, you can leave "SYSTEMIP".
- Do you wish to join an existing region to Udin estate (yes / no)? [Yes]: This is used to attach the newly created region to an existing state. If there is a previously created state which wants to join the region you should write its name, or on the contrary, type " no "and proceed to create one.

The parameters that are required to create one are:
- Estate name [MyEstate]: alphanumeric character. Any

name.

The following data creates a new user on the server and will be the first user to access it:
- Estate owner first name [Test]: alphanumeric character. The name to use for the estate owner.
- Estate owner last name [User]: alphanumeric character. The name to be used for the estate owner.
- Password: alphanumeric character. User Password created.
- Email [ ]: alphanumeric character. Email of the newly created user.

Now we have a region, a state and a user. So, we can run the server and it can be accessed from any viewer. These parameters can be modified at any time by accessing the Regions.ini located in the bin/Regions directory within OpenSim. Text file appears as shown in Figure 2.

Fig. 2. View of Regions with location, internal address, port and the name UDIN server, Regions ID it is compose for a hexadecimal number unique in the world.

C. OpenSim settings for different network topologies.

OpenSim is a simulator that runs on a server, ie it is not a local program as a video game in which the user starts a process in his computer and it ends when he finishes using it. The situation resembles that of an interactive website in which the user makes requests to the server and starts or finishes sessions, accesses databases or performs specific actions in which processing occurs on the server (which may be in another location or on the same computer from which you connect the client). If you have a server, you must have a client with the necessary application to access its services. In case you have OpenSim Hippo viewers, like OSV, RealXtend or just the Second Life Viewer, they allow users to connect to servers running the simulator. Because the client and server can be located on the same network or on separate networks, we'll go on providing a brief explanation of the basic configurations of the above parameters, so that clients are able to access to servers.

Depending on the topology, we can have multiple address mappings (static or dynamic –depending on if the server has DHCP enabled or not-)

Changes on the server are made in the file Regions.ini, located in the bin/Regions folder within OpenSim. The only parameters that must be modified to access are "ExternalHostname" and "Internal Address".

It is possible to change the client directly off the display. The only parameter important to change is Login URI (the socket through which you access the server), it also allows viewers to save configurations and to assign names and other descriptive parameters that are not of interest here. Here are some settings for the server and the client, and network topologies.

- *Topology 1: Client and server are hosted on the same computer.*

The client and server configuration shown in Table I:

TABLE I
SETTING PARAMETERS FOR TOPOLOGY 1

Client	InternalAddress	0.0.0.0
	ExternalHostName	SYSTEMIP
Server	Login Uri	http://localhost:port

The network topology is shown in Figure 3.

Fig. 3. Topology 1, this computer hosts server and client software, the virtual worlds development builder is inside.

- *Topology 2: Client and server on the same network.*

In this case we have the client and server using the same network serving different IP addresses (usually within a network with private addresses). The configuration is shown in Table II:

TABLE II
SETTING PARAMETERS FOR TOPOLOGY 2

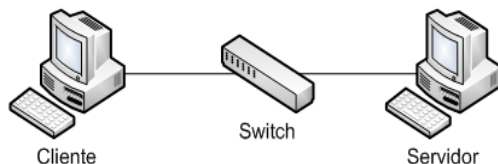|		
Cliente	InternalAddress	0.0.0.0
	ExternalHostName	SYSTEMIP
Servidor	Login Uri	http:// public_address:server

The network topology is shown in figure 3.

Fig. 4.. Topology 2, the client and server using the same network and serving different IP addresses

- *Topology 3: Client and server connected via the Internet, where the server has a public address.*

This configuration occurs when the server is connected to the internet through a broadband modem provided by an ISP and obtains a public address. The server configuration is the same as in the previous case. On the client, we can use this server's public address as part of the login URI. Figure 5 shows this network topology.
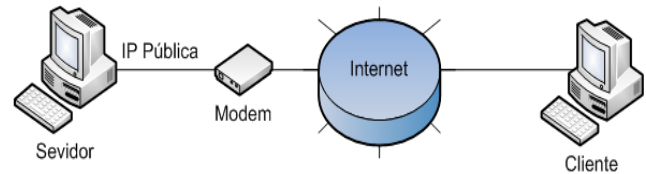

Fig. 5. Topology 3 Client and server connected via the Internet, where the server has a public address over the Web.

- *Topology 4: Client and server connected via the Internet, where the server has a private address.*
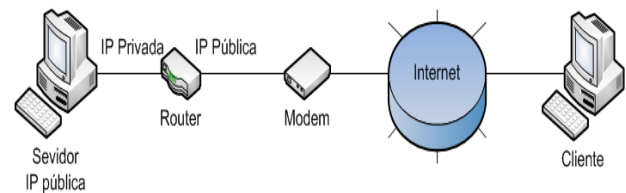

Fig. 6. Topology 4 Client and server connected via the Internet.

This configuration occurs when the server is connected to internet through a router connected to the ISP modem and therefore has a private address, which is translated by the router (NAT) to get out to the Internet. In this case, in addition to the above settings, we should enable port forwarding in the router for transport protocols TCP and UDP.

TABLE III
SETTING PARAMETERS FOR TOPOLOGY 4

Cliente	InternalAddress	private _address
	ExternalHostName	public_ address
Servidor	Login Uri	http:// public_address:port

To allow the client to connect to the server in the latter topology is also necessary to make port forwarding in the router. In current routers this function is usually called "virtual server" and is used to specify which ports redirect the public address to a private address.

D. Construction of objects in OpenSim

Creating and modifying objects in OpenSim is the basis for the construction of a metaverse. The first thing you should do is increasing the size of the island, this is accomplished using the "terrain fill" command long with the number of the desired size. The region may have a maximum size in the grid, because if set to a larger size, it would overlap with other regions in grid mode.

Different tools within OpenSim are:

- Vision (ctrl + 1) is used to change the angle of view, the approach and orbit of the camera.
- Move (Ctrl + 2): With this option you can move the different objects in the region.
- Edit (Ctrl + 3): this is one of the most important tools; this tool can easily modify position, rotation and size of objects.
- Create (ctrl + 4): This tool creates different primitive objects like cubes, cylinders, cones, spheres, prisms, toroids and others.
- Ground (ctrl + 5): With this tool you can raise, level, smooth down or reset the selected field. Using this tool you can also change the properties to specific "plots", including the type of texture that will be used to play media.
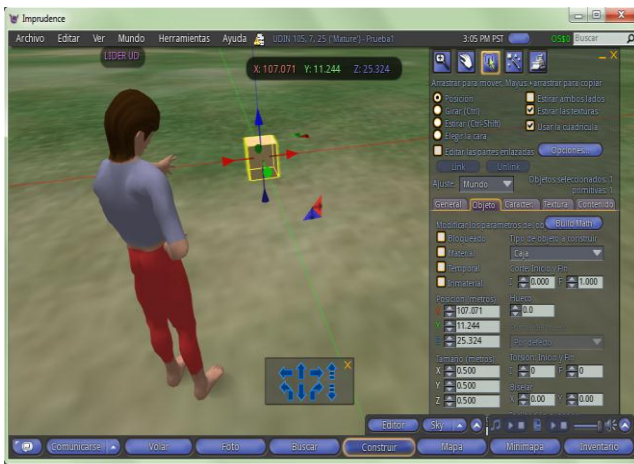

Fig. 7. OpenSim viewer, basic tools menus and tabs for designing objects in the virtual worlds.

When you open any of the aforementioned tools, you can make specific modifications to the objects just created, using the tabs at the bottom of the popup window.

The tabs available in the tools menu are:
- General: here you can make a description and edit the permissions and ownership of objects, and define the action performed by default when an avatar touches it.
- Purpose: In this tab you can set "physical" properties of the object in the metaverse, such as position, rotation, size, or even if the object is material, ghost, etc. The detailed description of the management of client tools for building an OpenSim metaverse are not the subject of this document.
- Features: edit the characteristics associated with the physics engine simulator. OpenSim accepts different physics engines that allow more realistic simulations of objects, but for the purpose of this project are not relevant.
- Texture: each object may also have different textures associated, which create more realistic environments and are also used for multimedia playback within the metaverse.

The complexity of building a metaverse starting from scratch using basic objects and the time it takes will depend on the user's ability using these tools. The detailed description of the metaverse construction is not the subject of this document.

IV. LINKING THE METAVERSE CREATED WITH MOODLE

If you have a Moodle platform installed on a server accessible to the client, this can be linked to the OpenSim metaverse created using Sloodle tools, so that users can make use of some of Moodle functions directly from within the metaverse. For this project we installed Moodle on the same machine hosting an OpenSim server, in order to perform some tests.

A. Installing Sloodle

The files required for Sloodle operation must be deployed on the server and on the user/client machine. The files of the server are available for download at the official project wiki. The decompressed folders must be copied into the following subfolders. It is important to remove any previous version of Sloodle directories installed in Moodle, before copying the new ones.
- Module (required): You must copy the "Sloodle" folder into the "mod" folder within Moodle.
- Block (Optional): You must copy the "sloodle_menu" folder into the "blocks" folder of Moodle.
- Plugin Sloodle object allocation (Optional): You must copy the "sloodleobjet" folder into the "mod/assignment/type" folder within Moodle.

The user files for OpenSim virtual objects are required to make use of the functionality of Moodle within the metaverse. This is called the Sloodle Set. Once downloaded, it must be loaded into the metaverse via the "load oar" command using the command line in the OpenSim server. Keep in mind that this loads a new region, so before you do so, it is advisable to save the region on which you are working.

Once SLOODLE oar has been loaded, the objects appear. Sloodle objects therefore must be stored in the user's inventory. With the above, the modules are installed and Sloodle is ready for use with OpenSim.

B. Moodle in OpenSim

Before you can access Moodle from OpenSim using a user with admin role, you must enter edit mode through the "Enable Editing" button located in the upper right of the course and proceed to click on the "Add activity" drop down menu and then the "SLOODLE Controller". This leads to a page designed to setup the data controller, of which we only need to supply the name, so after filling this field, so you can click Save and return to course.

This controller handles all communications between Moodle and OpenSim and is therefore necessary to use any other Sloodle tools.

Some of the Sloodle tools, such as "Presenter", must have a Moodle driver object that is added the same way as the main

controller. These tools added from Moodle page are required for objects that are added to Sloodle metaverse and they have some functionality, otherwise you cannot access the contents of the Moodle database. Other tools such as the "Sloodle Choice" or "Quiz Chair" can interact directly with native Moodle tools as "Choice" or "Quiz" respectively.

In this case the example will be a "Presenter" because you need to make video presentations in the metaverse. The detailed explanation of each one of the Sloodle tools is not the purpose of this paper and is left to the reader. As previously stated, the driver is first added to Moodle using "Add Activity" and then "Presenter". At this point, you must add the metaverse object. All Sloodle client tools can be added through the metaverse's "Sloodle Set", added above the metaverse and kept in inventory.

Before performing any task in Moodle using OpenSim, you must link the SLOODLE set located on the client host with Sloodle Controller located on the server. The steps for doing this are:

- Step 1: Click the SLOODLE Set
- Step 2: Write in Moodle site address, for example http://localhost/moodle or http://200.56.46.103:54661/moodle.
- Step 3: A dialog box appears in the viewfinder. Continue clicking on "Go to page".
- Step 4: To authorize objects Sloodle opens a browser which in turn shows the viewer . It should come with admin role, or shows the course in which you added the controller . Select "authorize previously added to Moodle", and post the form, after which you can close the browser.
- This should be done only once for the main controller Sloodle set, and once for each object you add to SLOODLE set in the metaverse.
- Step 5: Finally you must click again on the Sloodle Set, and then click Download configuration, so that the object has authorized access to the specified Moodle platform.

At this point you have a connection between OpenSim and Moodle but still have no functionality. Functions are added by SLOODLE set objects. The steps to add a Sloodle tool are:

- Step 1: Click the Sloodle set.
- Step 2: Click "Rez an object"
- Step 3: Click the object number to be added to the metaverse, in this case "7" representing the Presenter.
- Step 4: Click the object and then "Go to page" to authorize the new object in Moodle, as explained above for the controller and then close the page.
- Step 5: Click on the object and then select "Download configuration"

If everything is ok, the object is linked to the Moodle platform. These changes are required in Moodle as Sloodle objects do not store information, all the information is in the Moodle database.

- Module (required): You must copy the "Sloodle" folder

into the "mod" folder within Moodle.
- Block (Optional): You must copy the "sloodle_menu" folder in the "blocks" folder within Moodle.
- Plugin Sloodle objects allocation (Optional).

V. LIBRARY DESIGN FOR VIRTUAL WORLDS

The use of digital libraries has changed over the years and even more with the advent of the Internet, from that moment the way knowledge is accessed really changed because in cyberspace there are countless sources of information: text, video and more. The change is so important that new generations of students prefer to search the web before going to a library; therefore arises the need for more innovative spaces that encourage children, youth and adults to enter virtual worlds and visit this new interaction spaces and new models of accessing knowledge. The ACL [12] make rules for Distance Learning Library Services, since 2000.

The fundamental basis for the creation of a virtual library are similar to a real repository but the interface is implemented by Sloodle, and therefore it must have intelligent access protocols. This is achieved using Web-based architectures, OWL, XML [13] [14], Ontologies, Semantic Services for RDF presentation [15] , data management, Grid repositories [16], databases compatible with XAMPP, connection to external sources of scientific data, social, geographic and more. All of them can be achieved by using Service Oriented Architectures (SOA) [17].
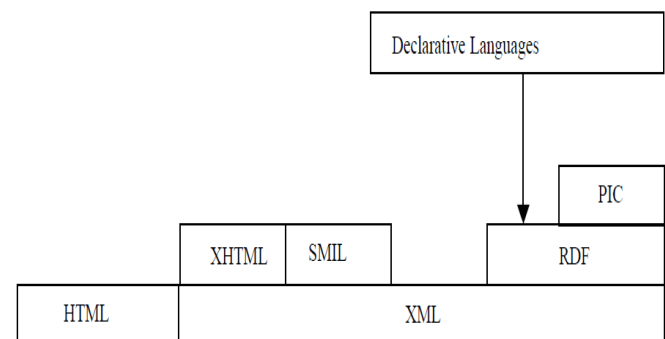
Fig 8. Language Layers on the Web, the Semantic Web, with machine processable information defined for their semantic content. Such semantic interoperability thanks of the W3C, XML/XML Schema and RDF/RDF Schema [13].
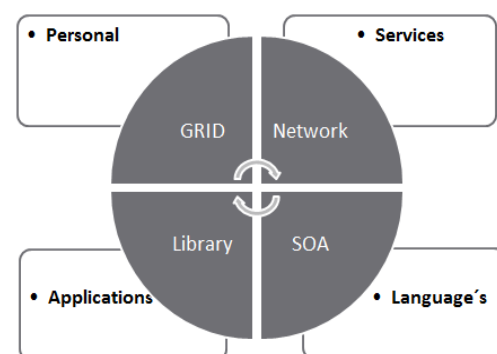
Fig. 8. Example using SOA for building virtual libraries over virtual worlds

One interesting and promising way of accessing the virtual Library at known places where we would like to promote it, is through publicly accessible NFC smart tags. In this scenario, when a user touches (or approaches) these tags with an NFC-enabled mobile phone, an event fires up an app which grabs the necessary URI that allows the user to access the virtual world.

This use of NFC technology can also act as a check-in method, or proof of attendance at known places where the Virtual Library is being promoted.

As seen, it is necessary to involve quite a lot of human resources, as researchers, Engineers, software developers, university students, research groups in the field of network architecture and broadband access to high speed Internet, Grid computing [18] and SOA services and a community of architects builders and 3D metaverse designers [22].

VI. RESULTS

The main result is a metaverse based on the engineering faculty of the "Francisco José de Caldas" Distrital University [19], hosted on a server accessible from the internet, and tested for Video Streaming [20] using OpenSim textures and mobile networks inside the University campus .

In Figure 10 you can see the photo of the real Engineering faculty building, and at the right, OpenSim built the building as proposed for the operation of a virtual 3D library. When the user logs in and reincarnates as an Avatar, is able to visit the site and see all kinds of information, all of the above accomplished by using Sloodle. In the future, we expect to further develop this initiative in collaboration with security staff trained to feed the virtual world.

Fig. 9. Metaverse example created in the Distrital University, with access to virtual library, Left real Building, Right virtual 3D Building in Opensim.

Also, there are objects -within the metaverse- linked to the Moodle platform installed on the same server.
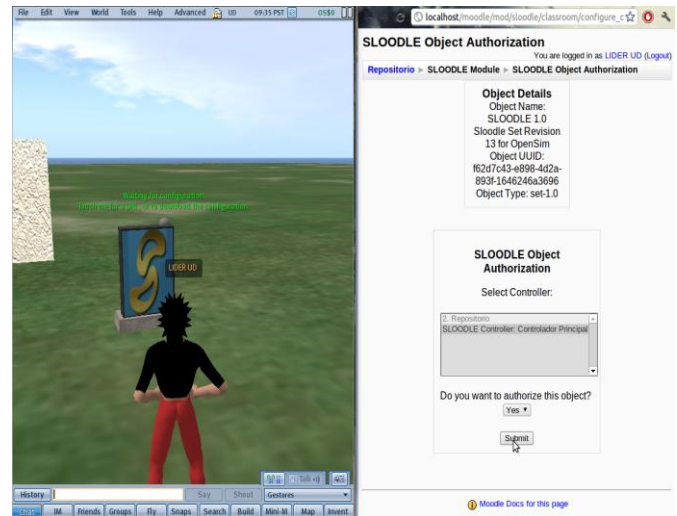

Fig 10. View window Sloodle controller with access from the metaverse.

VII. CONCLUSION

These kind of metaverses may target the educational virtual environments and the dissemination of scientific material, even in simulation labs, thanks to the programming tools here presented, which allow the control of the characteristics of the objects in these metaverses. Different research groups of the Distrital University are working with free tools like OpenSim and finding their application in specific fields of knowledge, and also other universities have already made significant progress: a good example of application is the development of 3D virtual libraries in these worlds.

One promising research area is the mobile broadband access to these virtual digital libraries, just making the knowledge ubiquitous, and special interest should be paid to recent proximity based interfaces such as NFC, which will be prevalent in mobile devices in the very near future, and a invaluable tool for check-ins as a gate between real and virtual worlds.

The use of virtual digital libraries help the environment as they save resources as printed paper, physical visits to sites, reduction in greenhouse gas emissions, electricity, time of travelling, encourages telecommuting from home and an endless resources that help to improve our world.

REFERENCES

[1] G. P. Fishwick, "An introduction to Opensimulator and virtual environment agent-based M&S applications," in Simulation Conference (WSC), Proceedings of the 2009 Winter, Dec. 2009, pp. 177 –183.64.

[2] Linden Research, Inc, Available: http://lindenlab.com

[3] M. Barbulescu, M. Marinescu, O. Grigoriu, G. Neculoiu, V. Sandulescu, and I. Halcu, "GNU, GPL in studying programs from the systems engineering field," in Roedunet International Conference (RoEduNet), 2011 10th, June 2011, pp. 1 –4.

[4] Hippo OpenSim Viewer, Available: http://mjmlabs.com/viewer

[5] RealXtend Viewer, Available: http://realxtend.org

[6] M. Pattal, Y. Li, and J. Zeng, "Web 3.0: A real personal web! More opportunities and more threats," in Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International, Conference on, Sept. 2009, pp. 125 –128.

[7] T. Kuroiwa and S. Bhalla, "Dynamic personalization for book recommendation system using web services and virtual library enhancements," in Computer and Information Technology, 2007. CIT 2007, 7th IEEE International Conference on, Oct. 2007, pp. 212 –217.

[8] M. Phankokkruad and K. Woraratpanya, "Web services for learning management systems: Communication architecture, in Communications (MICC), 2009 IEEE 9th Malaysia International Conference on, December. 2009, pp. 403 –408.

[9] M. Di Giacomo, "MySQL: Lessons learned on a digital library," Software, IEEE, vol. 22, no. 3, pp. 10 – 13, May-Jun 2005.

[10] S. Spaeth, M. Stuermer, S. Haefliger, and G. Von Krogh, "Sampling in open source software development: The case for using the debian gnu/linux distribution," in System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, Jan. 2007, p. 166.

[11] D. D. Dvorski, "Installing, configuring, and developing with Xampp," March 2007.

[12] Content and Design of Academic Library Web Sites for Distance Learners: An Analysis of ARL Libraries, Association of College & Research Libraries. Guidelines for distance learning library, December 2001.

[13] Hannes Bohring, Soren Auer, "Mapping XML to OWL Ontologies" June (2005) Available:
 http://www.informatik.unileipzig.de/~auer/publication/xml2owl.pdf.

[14] Brewster, C. O. (2004). "Knowledge Representation with Ontologies: The Present and Future". IEEE, 72-81.

[15] Simons, Gary F., William D. Lewis, Scott Farrar, D. Terence Langendoen, Brian Fitzsimons and Hector Gonzalez. 2004b. The semantics of markup: Mapping legacy markup schemas to a common semantics. Proceedings of 4th Workshop on NLP and XML: RDF/RDFS and OWL in Language Technology. Association for Computational Linguistics, Barcelona, Spain, July 2004. Online: http://emeld.org/documents/SOMFinal1col.pdf

[16] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke. "A resource management architecture for metacomputing systems". In IPPS/SPDP '98. Workshop on Job Scheduling Strategies for Parallel Processing.

[17] R. Fracchia, G. Jeney, M. Mazzotti, E. Piri. (2011). Optimization of Multimedia over wireless IP links via X-layer design" University of Bologna.

[18] Niaye NM, et. al.. "Performance comparison of hierarchical checkpoint protocols grid computing". International Journal of Interactive Multimedia and Artificial Intelligence. 2012, v 1, n 5, (pp 46-53).

[19] M. Erdmann, D. Fensel, I. Horrocks , M. Klein, S. Melnik. "The Semantic Web on the respective Roles of XML and RDF". IEEE, In Conference publication in the Third Workshop on Workflows in Support of Large-Scale Science, On 17-17 Nov. 2008, pp 1-10.

[20] Francisco José de Caldas Distrital University 1948 in Bogotá Colombia, Available: http://udistrital.edu.co

[21] Julian Camargo, Roberto Ferro,. (2009). "Research Laboratory for Development Electronic and Networks" LIDER-UD". Available: http://201.234.78.173:8080/gruplac/jsp/visualiza/visualizagr.jsp?nro=00 000000005315.

[22] Ndiaye, N. M., P. Sens, and O. Thiare, "Performance comparison of hierarchical checkpoint protocols grid computing", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 1, issue

Special Issue on Distributed Computing and Artificial Intelligence, no. 5, pp. 46-53, 06/2012 DOI: 10.9781/ijimai.2012.156

M.Sc. Roberto Ferro Escobar, Electronic Engineer, M.Sc. in Networks 2004, became a Member of IEEE in 1999, a Senior Member in 2011, teaching at the "Francisco José de Caldas" Distrital University since 2006 in Bogotá Colombia, ex-manager of the Research Network of Advanced Technology "RITA", former manager program of Electronic Engineering 2009-2011, is currently developing his Ph.D thesis on Computer Engineering from the Universidad Pontifical of Salamanca in Madrid Spain.

Ph.D. Rubén González Crespo, Ph.D. in Computer Engineering from the Pontifical University of Salamanca in Madrid Spain, lecturer in Sciences and Engineering in the same university.

M.Sc. Eng. Sergio Ríos Aguilar. MSc in Telecommunication Systems, Computer Engineer, and PhD Candidate, lecturer in Sciences and Engineering at Pontifical University of Salamanca in Madrid, Spain Researcher on advanced mobile services.

Eng. Nicolas Torres Electronic Engineer became a Member of IEEE in 2010, student at the "Francisco José de Caldas" Distrital University in Bogotá Colombia 2006, Active member of LIDER research group.