Expert Systems with Applications 40 (2013) 7381–7390

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Use of ARIMA mathematical analysis to model the implementation of expert system courses by means of free software OpenSim and Sloodle platforms in virtual university campuses ☆

Rubén González Crespo [a,*], Roberto Ferro Escobar [b], Luis Joyanes Aguilar [a], Sandra Velazco [b], Andrés G. Castillo Sanz [a]

[a] Pontifical University of Salamanca, Madrid, Spain
[b] Francisco José de Caldas, District University, Bogotá, Colombia

ARTICLE INFO

Keywords:
Expert systems
Artificial intelligence
OpenSim
Sloodle
ARIMA

ABSTRACT

This paper describes the implementation of a virtual World based in GNU OpenSimulator. This program offers a great variety of Web 3.0 ways of work, as it makes possible to visit web sites using avatars created for that purpose. The Universities should be familiar with the creation of new metaverses. That is the reason why a new basic methodology for the creation of a course on expert systems within a metaverse in a virtual campus for e-learning. Besides the creation of a repository or island, it is necessary to make measurements of the performance of the server dedicated to host the system when the number of users of the application grows. In order to forecast the behavior of such servers, ARIMA based time series are used. The auto-correlogrames obtained are analyzed to formulate a statistical model, as close to reality as possible.

1. Introduction

The e-learning project of the "Francisco José de Caldas" District University in Bogotá (Colombia) is currently on the first phase. The objective of the first phase is to learn about the use of virtual tools and their applications to carry out e-learning projects. Another goal is to overcome the difficulties associated with the use of appropriate visual interfaces and their adequate implementation. There is not much information at this moment about disadvantages that could impede the construction of a proper interface with the database and the LMS. The challenge lies in how to tackle new ways of presenting and visualizing contents, which it is something under much discussion nowadays, regarding which are the paths to keep us away from the static web that only allows the reading of documents, books and publications on cloud repositories. There are many teachers and institutions who claim to have better ways of accessing those new sources of information, in particular for the development of science courses on applied Artificial Intelligence.

Thanks to globalization, improved internet speeds and global interconnection between networks, it is now feasible to access learning contents without the need of physical traveling, as the students can make use of these ITC resources no matter where they are. This shows the evolution from a static Web 2.0 to a dynamic Web 3.0. This change is made possible also by the development of simulators of virtual worlds such as Open Simulator (Fishwick, 2009). This simulator is created under the guidelines of SecondLife (by Linden Labs Linden Research) and it was released in 2007 and placed under the GPL (Barbulescu et al., 2011). This simulator is characterized by a modular structure written in C# and it supports multiple viewers such as the viewer Hippo OpenSim Viewer; RealXtend Viewer and the same SecondLife official viewer.

Open Simulator complies with the client–server model. In the server the interconnected islands or rectangular regions are created, and the viewer is installed on the client. The viewer has similar functions as Mozilla Firefox or Google Chrome but they are used to access virtual worlds. The dynamism of Web 3.0 (Pattal, Li, & Zeng, 2009) combines artificial intelligence techniques, geospatial mobility, interactivity, multimedia 3D, and the transformation towards a semantic database capable of learning.

The combination of access to courses of expert systems and artificial intelligence on virtual worlds built with GNU tools besides the use of a statistical package is an excellent way to teach complex topics to students interested in them.

☆ Paper submitted on April 4, 2013. This work is part of the developments about e-leaning free courses made under the model of Scientific Knowledge Management Research Network of Advanced Technology "RITA" in the District University "Francisco José de Caldas" assigned to the Center of Scientific Research and Development.

* Corresponding author. Tel.: +34915141700.
E-mail addresses: ruben.gonzalez@upsam.es (R. González Crespo), rferro@udistrital.edu.co (R.F. Escobar), andres.castillo@upsam.es (A.G. Castillo Sanz).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.06.054

Artificial intelligence is a science that deals with the design of computer systems that are capable of displaying characteristics that we associate with intelligence in human behavior: language comprehension, reasoning, learning and problem solving, among others. This has led to the emergence of numerous support tools for occupations, tasks and activities in a variety of areas. The virtual worlds have been no exception to that with the application of AI to many aspects of the benefits of digital virtual libraries, such as research, special papers, and knowledge management. Similarly, the vast amount of information emanating from student thesis and their teachers lends itself to input in databases for the extraction of embedded knowledge from the perspective of different forms of AI or its affiliated areas. This approach is contextualized within ITC development in Computer Sciences, some examples of which are systems databases and Storage Area Network SAN (Tate, Bernasconi, Mescher, & Scholten, 2003).

2. 3D Free tools to create a course in expert systems

2.1. OpenSim 3D modeler

OpenSim is a 3D application server that uses the same standards as SecondLife to communicate with their users. It emerges from the analysis of the structure of the source code viewer of SecondLife, released in early 2007 by Linden Labs under GPL, reverse engineered in order to create a 3D application server. It is characterized by being free software, it has a modular structure written in C# and it supports multiple viewers. To date OpenSim is on version 0.7.

One of the biggest potentials of OpenSim is the ease with which a university can customize its development, design management systems, and integrate databases of LMS (Learning Management Systems) (Phankokkruad & Woraratpanya, 2009) with OpenSim database, so that content can be created to suit the needs and methodologies of each institution (see Fig. 1).

OpenSim is used for designing, creating and easily implement 3D virtual worlds by many programmers around the world. Content targeted to different areas can be produced with this tool, such as e-learning, virtual tours, and simulations, amongst others. Moreover, being a free software tool, it enables low cost development and allows developers around the world to contribute to the development of new features.

2.2. Moodle for a Web courses

Moodle is an acronym for "Modular Object-Oriented Dynamic Learning Environment." It is a software package for creating courses and web sites based on open source and developed in PHP. It is designed to work on different database engines specially MySQL (Di Giacomo, 2005), with the objective to help educators create and easily manage online learning communities. Moodle is designed to foster the building of knowledge for the student.

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

Its first version was published in August 2002 and to date the current version is 2.2. Moodle currently has over 4.780,000 courses offered through 56,361 registered sites in 215 countries, including Colombia which is the eighth country in registered courses (see Table 1).

2.3. Sloodle project

The project SLOODLE (Simulation Linked Object Oriented Dynamic Learning Environment) (Spaeth, Stuermer, Haefliger, & Von Krogh, 2007) is a success code that integrates multi-user virtual environments like Second Life and OpenSim with the learning management platform Moodle. Sloodle provides a wide range of tools (some still in development) that enable virtual learning worlds and immersive education (Getchell, Miller, Nicoll, Sweetman, & Allison, 2010; Lala & Nishida, 2011). Such tools are fully integrated with a learning management system based on the web. It is already used by thousands of educators and students worldwide.

2.4. Ubuntu operating server

Ubuntu is an operating system that uses a Linux kernel and it is based on Debian (Spaeth et al., 2007). It is maintained by the British firm Canonical and a 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 50% preference among Linux distributions.

For the development of this project it should be noted that Linux is characterized as an operating system designed from the beginning as multithreaded, multiuser, and it manages the permissions for each user on processes and files from access by other users in a very secure way. Ubuntu comes by default, with the "root" user disabled, so that the system repeatedly would ask the user's password to install any components and it is also known to be very careful with permissions at the time of modifying files and folders.

3. OpenSim and virtual worlds

To create a virtual course the following steps are to be carried out (Torres & Uribe, 2012):

3.1. Installing XAMPP database

It is necessary to have a web server to accommodate the contents of Moodle, and it is advisable to configure MySQL as the default database engine in the OpenSim server. We would need to use the XAMPP (Dvorski, 2007) Apache distribution which includes the installation of MySQL. Linux installation files for XAMPP are available on the official website of "Apache Friends". Once the file has been downloaded, it should be decompressed under the "opt" file system. Finally the server must be started using the command "/opt/lampp/lampp start".

One of the utilities of XAMPP is the administration tool for MySQL databases from php: phpMyAdmin, which can be accessed from the home page of the server by typing http://localhost in a browser.

The admin page does not have an enabled access control by default, so anyone with access to the server can change all the

Table 1
List of used tools with their descriptions.

Tool	Description
OpenSimulator	This software provides a platform on which the developer's community can build virtual worlds that can be exchanged, learned, tested, analyzed, and improved through multi-institutional collaboration. The underlying software is written in ANSI C++, and the graphical user interface (GUI) is written in Java.
PHP	PHP is a server-side scripting language designed for web development Interface for OpenSimulator it allows grid citizens to create user accounts to access the metaverse grid. Grid owners can also manage all users for the grid, with a very powerful CMS system included.
MySQL	It is a GNU open source relational database management system (RDBMS). It is used to set StorageProvider and ConnectionString so that OpenSimulator can connect to a database.
Sloodle	It is an open source tool used to create a LMS base while developing at the same time their own supported solution for avatar classroom. It is similar to Moodle and allows the educator to set up individual learning exercises dynamically created and deleted on-demand for each student.
Hippo viewer	The Hippo OpenSim Viewer is a modified Second Life viewer, to access a virtual world. It allows the building up to a height of 10,000 m, scaling primes up to $256 \times 256 \times 256$ m and other exciting features. It is run in Windows and Linux clients.
Ubuntu server	It is an open source operating system for 3D virtual world server. It allows multiple avatar users to custom build a virtual environment, and to interact with each other in real time.

parameters. It is therefore recommended to set up access passwords. The XAMPP documentation describes how to do this.

3.2. Installing the metaverse world with OpenSim

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

OpenSim was developed to run on the Microsoft dot NET framework and much of the code and scripts used in the metaverse are based on C#, so in order to install it on Linux it is required to install the Mono framework, which has the necessary libraries to compile those languages. The files needed to install Mono on Linux are in their official download page, they can be downloaded to the terminal with the installation commands too (e.g. Ubuntu: sudo apt-get install mono-complete). After having installed mono, the OpenSim file should be unzipped using the terminal. To start the server, the command "mono OpenSim.exe" should be entered with root permissions.

If there are not any compatibility issues, the server should start. This loads all the necessary dependencies to the metaverse one by one. If this is the first time OpenSim is run on the computer, and if there is not any metaverse already created, then the simulator would request the creation of one, asking the information listed below:

- New region name [ ]: alphanumeric characters. Name that will be given to the region. Enter the desired name for the region.
- RegionUUID [UUID_by_default]: UUID. Unique identifier for this region. OpenSim proposes a default, so you could just press enter unless you prefer a specific UUID.
- Region Location [1000.1000]: *x*, *y* coordinates. It is the desired coordinate point for the region. Mode must be specified for non-grid overlay on existing regions. Otherwise the default one can be accepted by pressing the enter key.
- InternalAddress [0.0.0.0]: Wildcard Mask (IP address). This address is actually a wildcard mask that allows the owner to restrict the range of addresses that can access the server. In most cases it should be left to 0.0.0.0 by pressing enter, so that all hosts can access it.
- InternalPort [5001]: # port (0-65535). Internal server port to be used for access. If port 5001 is not being used by other application, pressing the Enter key means accepting the default.
- AllowAlternatePorts: True or False. Function still in the experimental stage. To be left false by pressing enter.
- ExternalHostname: IP address. This is the address trough which clients will access the server, and it should take into account some considerations that are discussed later. To start it could be left as "SYSTEMIP".
- Do you wish to join an existing region to Udin state (yes/no)? [Yes]: This is used to attach the newly created region to an existing state, if there is a previously created state to which the administrator wants to join the region, and then the name you should be written. If there are not any states then by typing "no" it proceeds to create one.

The parameters that are required to create one are:

- State name [MyState]: alphanumeric characters. Any name.

The following parameters should be remembered so that later on a new user can be created on the server, who will be the first user to access the state:

- State owner first name [Test]: alphanumeric characters. The name to be used for the state owner.
- State owner last name [User]: alphanumeric characters. The surname to be used for the estate owner.
- Password: alphanumeric characters. User Password created.
- Email [ ]: alphanumeric characters. Email user created.

So now we have a region, a state and a user. With this ready, the server can be started. It can be accessed from any viewer. These parameters can be modified at any time by accessing the Regions.ini file located in the /bin/Regions folder of OpenSim.

3.3. Building objects into OpenSim

Creating and modifying objects in OpenSim are the basic operations for the construction of the metaverse. The first thing to do is to increase the size of the island. For this the command "terrain fill" and the desired number should be used. The region may have the maximum size of the grid, because if it is set to a larger size it would overlap with other regions in grid mode.

Different tools inside OpenSim are (Gonzalez-Crespo, Rios-Aguilar, Ferro-Escobar, & Torres, 2012):

- Vision (Ctrl+1) is used to change the angle of view and the zoom of the camera.
- Move (Ctrl+2): With this option the different objects in the region can be moved.
- Edit (Ctrl+3): This is one of the most important tools; this tool changes easily the position, rotation and size of objects.

- Create (Ctrl+4): This tool creates different basic objects like cubes, cylinders, cones, spheres, prisms, toroid and others.
- Ground (Ctrl+5): With this tool the user can raise, level, smooth down or reset the selected field. Using this tool he or she can also change the properties to specific ''plots'', including the type of texture that will be used to play the media (see Fig. 2).

When any of the above tools is opened, different tabs will appear at the bottom of the pop up window in which the user can make specific modifications in the created objects.

The tabs available in the tools menu are:

- General: Here the owner can make a description and edit the permissions and ownership of objects, and define the action performed by default when an avatar touches it.
- Object: In this tab the user can set ''physical'' properties of the object in the metaverse, such as position, rotation, size, or even if the object is material, ghost, if it is locked, etc. For some basic objects he can set additional options such as cross sections, start and end cuts, define whether it is or not hollow, and its twist, among others is. The detailed description of the management of client tools for building in OpenSim metaverse are not the subject of this document.
- Features: It allows editing the characteristics associated with the physics engine simulator. OpenSim accepts different physics engines that allow more realistic simulations of objects, nevertheless for the purpose of this project they are not relevant.
- Texture: Each object may also have different textures, which create more realistic environments and it is also used for playback of multimedia within the metaverse.

The building of complex metaverse starts from basic objects and the time it takes to complete it will depend on the user ability to use the tools abovementioned.

3.4. Connecting the new metaverse with Sloodle

Once the administrator has the Moodle software and the Sloodle tools running, its functions can be used directly into the metaverse. For this course we installed Moodle on the same server of OpenSim, and then we linked the Sloodle set located on the client host with the Sloodle Controller located in the server. The steps to carry out this are as follows:

- Step 1: Click on the Sloodle set.
- Step 2: Write in the chat the Moodle site address, for example http://localhost/moodle or http://192.168.1.100:5001/moodle.
- Step 3: A dialog box appears in the viewfinder. Then click on ''Go to page''.

Fig. 2. OpenSim viewer and basic tools menu and tabs for designing courses in the virtual worlds.

- Step 4: To authorize Sloodle objects, a browser should be opened, which could be the one included in the viewer or the default browser of the operating system. The owner should log in with admin role or teacher of the course in which the controller was added, and then she should authorize it by selecting the previously controller added to Moodle, and then she should send the form. After this she could close the browser. This should be done only once for the main controller of the Sloodle set, and once for each object added to the metaverse.
- Step 5: Finally the adminsitrator should click again on the Sloodle Set, and then click on Download configuration, so that the object that was authorized may have access to the specified Moodle platform (see Fig. 3).

4. Why to teach an expert systems 3D virtual course?

There is still much work to be done about the programming of artificial intelligence, and the development of robots and smart machines. There is a great need of algorithms and software related to this important area of knowledge. That is an important reason for the universities to offfer courses that promote the study of those techniques and other topics regarding autonomous systems that could be capable of reasoning similarly to the human brain (Giarratano & Ricley, 1998).

4.1. Benefits of using OpenSim 3D virtual worlds in educational courses

- The teachers may cooperate with students on a less formal basis, and students can choose an individual learning program and establish cooperation with other students.

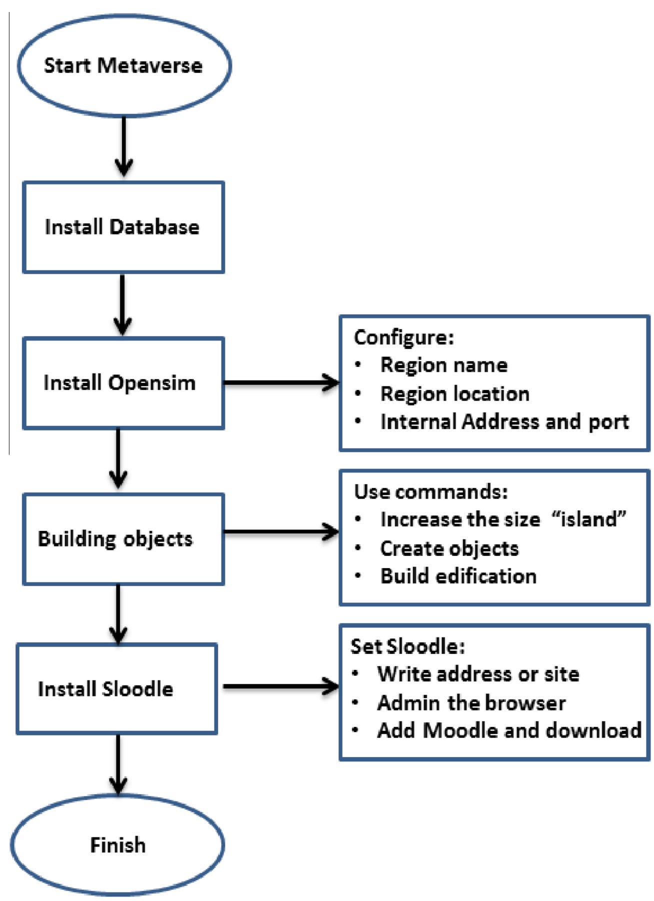

Fig. 3. Flow-chart to create a virtual course with Sloodle in the OpenSim.

- The course can be followed many times without having to be created again and again.
- Students can create avatars according to their taste and with the features they feel like.
- The course can create graphical objects that simulate the behavior of a hypothetical system with artificial intelligence.
- Virtual worlds have similar tools as the Web 2.0 for creating, sharing and transfer of new knowledge by using chats, forums, wiki, e-mail, video conference and other tools.
- It promotes the use of ICT to promote the grow of new developments in the area of computer and telecommunications systems.

4.2. The expert system course contains the following chapters Buckley and Eslami (2002), Russell and Norvig (2002)

Chapter 1 Introduction:

- Knowledge-acquisition interface
- User interface
- Knowledge base
- Inference engine

Knowledge-acquisition interface

The knowledge-acquisition interface controls how the expert and knowledge engineer interact with the program to incorporate knowledge into the knowledge base. It includes features to assist experts in expressing their knowledge in a form suitable for reasoning by the computer.

User interface

The user interface is the part of the program that interacts with the user. It prompts the user for information required to solve a problem. It displays conclusions and explains its reasoning.

Knowledge base

The knowledge base consists of specific pieces of knowledge about some substantive domain. A knowledge base differs from a data base in that the knowledge base includes both explicit knowledge and implicit knowledge. The knowledge bases can contain many different types of knowledge together with the process of acquiring knowledge into the knowledge base.

Inference engine

The inference engine uses general rules of inference to reason from the contents of the knowledge base and draw conclusions which are not already explicitly stated, but can be inferred from the knowledge base. Inference engines are capable of symbolic reasoning, not just mathematical reasoning. Hence, they expand the scope of fruitful applications of computer programs.

The specific forms of inference permitted by different inference engines vary, depending on several factors, including the knowledge representation strategies employed by the expert system.

The course on expert systems presents many visions about knowledge management:

- Explicit knowledge
- Implicit knowledge
- Domain knowledge
- Common sense or world knowledge
- Heuristics
- Algorithms knowledge
- Procedural knowledge
- Declarative or semantic knowledge
- Public knowledge
- Private knowledge
- Shallow knowledge
- Deep knowledge
- Meta-knowledge

Chapter 2: Understanding the expert systems:

- Sample applications.
- Tools for development expert systems.
- Software for building, maintaining, and compiling rule sets.
- Potential benefits to use expert systems.
- Evolution and new applications.

Chapter 3: Rules for Reasoning:

- Premises to conclusions.
  - o IF, Then.

Fig. 4. Overview about the welcome screen to the course of expert systems in the OpeSim 3D virtual world created.

Table 2
Number of user, CPU percent utilization and real speed variation tested in a 3D virtual world course.

User	CPU utilization (%)	Variation (%)	Speed CPU (MHz)	Variation (MHz)
1	4.2	0	162.3	0
2	6.4	2.2	208.2	45.9
3	10.9	4.5	297.2	89
4	14	3.1	378.7	81.5
5	16	2	427.4	48.7
6	21.3	2.5	587.79	68.25
7	24.6	3.3	668.37	80.58
8	25.9	1.3	736.98	68.61
9	28.3	2.4	809.73	72.75
10	31.2	2.9	888.75	79.02
11	33.7	2.5	966.21	77.46
12	36.3	2.6	1011.23	45.02
13	37.1	0.8	1080.33	69.1
14	39.5	2.4	1134.27	53.94
15	42.2	2.7	1201.97	67.7
16	45.1	2.9	1284.66	82.69
17	47.4	2.3	1362.65	80.99
18	49.3	1.9	1409.02	43.37
19	52.4	3.1	1455.49	46.47
20	55.2	2.8	1536.55	81.06
21	58.1	2.9	1599.3	62.75
22	60.5	2.4	1678.4	79.1
23	63.2	2.7	1759.37	80.97
24	65.6	2.4	1812.78	53.41
25	67.3	1.7	1900.47	87.79
Average	36.712	2.427	1033.872	66.853

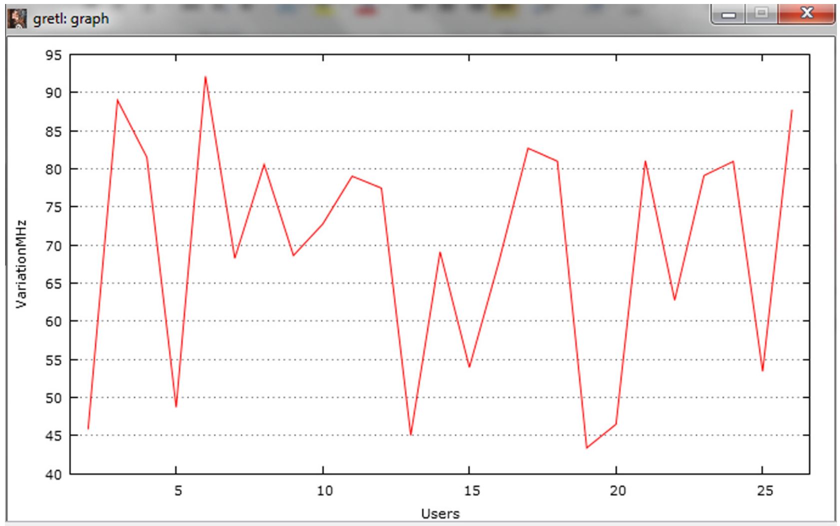

Fig. 5. Time series original function in GRETL.

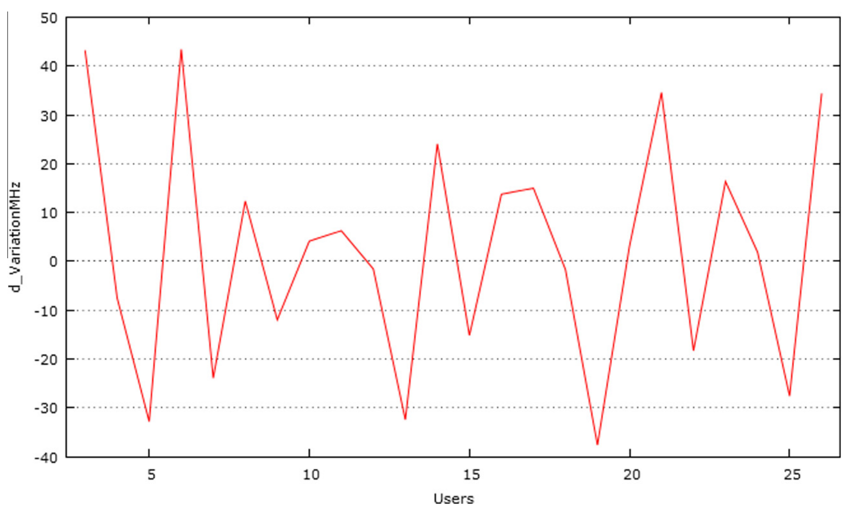

Fig. 6. Plot time series function and his first derivate.

  - o Boolean.
  - o Conjunction, disjunction.
  - o Exclusive OR.
  - o Input actions.
- Managing Rules.
- Reasoning with Rules.

Chapter 4: Introduction to other applications:

- Data mining.
- Fuzzy logic.
- Simulation models.
- Neural networks.

Chapter 5: On line problems:

- Solved problems.
- Exercises proposed.
- Sharing new knowledge.
- Discussions and feedback.
- Course evaluation.
- New applied research.
- References.
- Conclusion.

5. Example of an expert system in a virtual world

The use of digital libraries changed over the years and the more so with the advent of internet. From that moment the way knowledge is accessed has changed because in cyberspace there are countless sources of information that comes in many formats like text, video and many more. The change is so important that new generations of students prefer to search the web before they go to a university; this arises the need for more innovative spaces that encourage these students, youth and adults to enter virtual worlds and visit these new courses of interaction and new models of access to knowledge (see Fig. 4).

The fundaments for the creation of effective hardware and software for expert systems begin by understanding how to use expert systems tools and then by applying the acquired knowledge to solve real problems. Such intelligent systems must be composed of sensors, actuators, and algorithms that make correct decisions based on rules and that use intelligent Web access protocols to search for knowledge. This is achieved by using Web-based

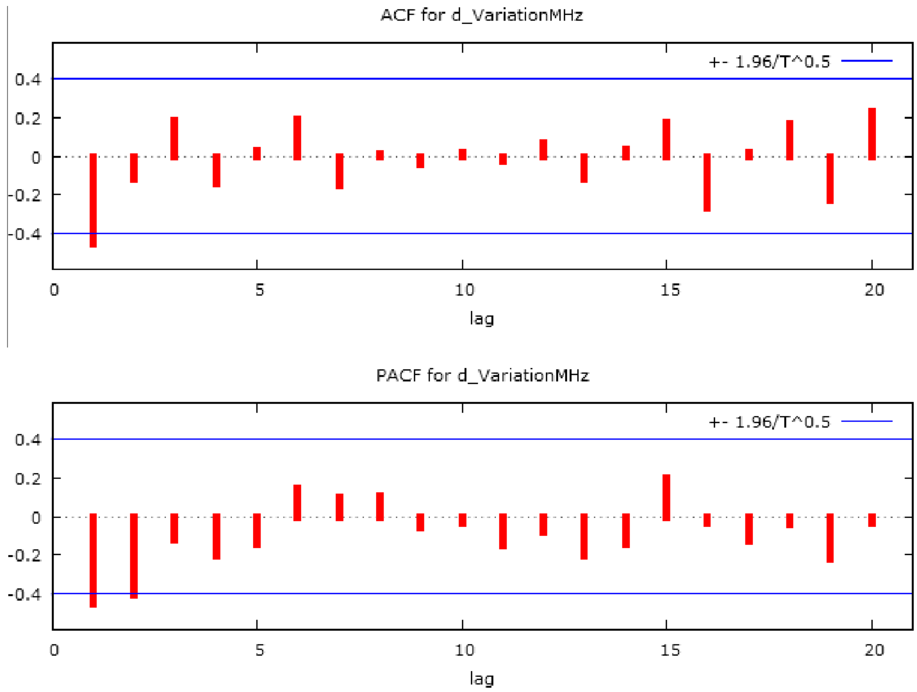

Fig. 7. Correlogram time series function ACP and PACF.

LAG	ACF		PACF		Q-stat.	[p-value]
1	-0.4498	**	-0.4498	**	5.4889	[0.019]
2	-0.1231		-0.4079	**	5.9185	[0.052]
3	0.1809		-0.1211		6.8906	[0.075]
4	-0.1415		-0.2016		7.5149	[0.111]
5	0.0285		-0.1412		7.5416	[0.183]
6	0.1922		0.1408		8.8224	[0.184]
7	-0.1519		0.0962		9.6694	[0.208]
8	0.0153		0.1035		9.6785	[0.288]
9	-0.0399		-0.0607		9.7448	[0.372]
10	0.0180		-0.0338		9.7592	[0.462]
11	-0.0303		-0.1512		9.8033	[0.548]
12	0.0689		-0.0784		10.0503	[0.612]
13	-0.1169		-0.2072		10.8257	[0.625]
14	0.0381		-0.1466		10.9162	[0.693]
15	0.1737		0.1982		13.0091	[0.602]
16	-0.2661		-0.0360		18.5322	[0.294]
17	0.0224		-0.1237		18.5768	[0.353]
18	0.1625		-0.0444		21.3233	[0.263]
19	-0.2297		-0.2207		27.9056	[0.085]
20	0.2257		-0.0345		35.8496	[0.016]

Fig. 8. ACF and PACF values.

```
Function evaluations: 41
Evaluations of gradient: 15

Model 1: ARIMA, using observations 4-26 (T = 23)
Estimated using Kalman filter (exact ML)
Dependent variable: (1-L) d_VariationMHz
Standard errors based on Hessian

             coefficient   std. error      z        p-value
  ----------------------------------------------------------
  const      -0.0297532    0.417397    -0.07128   0.9432
  phi_1      -0.525790     0.196744    -2.672     0.0075     ***
  theta_1    -1.00000      0.117319    -8.524     1.54e-017  ***

Mean dependent var  -0.383478   S.D. dependent var   40.12179
Mean of innovations -5.095709   S.D. of innovations  20.58810
Log-likelihood      -104.3628   Akaike criterion     216.7255
Schwarz criterion    221.2675   Hannan-Quinn         217.8678

                        Real  Imaginary    Modulus  Frequency
  -----------------------------------------------------------
  AR
    Root  1          -1.9019     0.0000     1.9019     0.5000
  MA
    Root  1           1.0000     0.0000     1.0000     0.0000
  -----------------------------------------------------------
```

Fig. 9. Model I ARIMA (1,1,1) function evaluation.

architectures, OWL, XML (Hannes & Soren, 2005; Isaza, Castillo, López, & Castillo, 2009, Ontologies (Brewster, 2004; Isaza, Castillo, López, Castillo, & López, 2010), Semantic Services of presentation RDF (Czajkoeski et al., 1997; Simons et al., 2004) type, data management, Grid repositories (Erdmann, Fensel, Horrocks, Klein, & Melnik, 2008), XAMPP compatible databases, connection to external sources of scientific data, social, geographic and more. All this can be arranged by using Service Oriented Architectures (SOA).

It is necessary to involve researchers in several scientific fields such as engineers, software developers, university students, research groups in network architecture and a community of architects and 3D designers of metaverse and courses similar to those exposed in this paper. It is important as well to have access to high speed Internet, grid computing and SOA services.

6. Traffic and experimental evaluation in ARIMA mathematical model

In this first phase of development the system performed measurements of traffic of 25 users who entered the 3D virtual world. The samples were extracted from the learning system and then they were included in the ARIMA (Cottrell & Lucchetti, 2008) mathematical model as a class of stochastic processes used to analyze time series according due to Box and Jenkins model:

$$\theta_p(L)(1-L)^d(y_t-\mu) = \varphi_q(L)u_t \tag{1}$$

The server used for testing had as operating system Linux CentOS 6.3, with hardware that was designed with a processing capacity of 16 cores each, with a processing speed of 2.93 GHz, Dell board, RAM of 8 × 2048 MB DDR3-1333 MHz and a DD-HDD space about 73 GB.

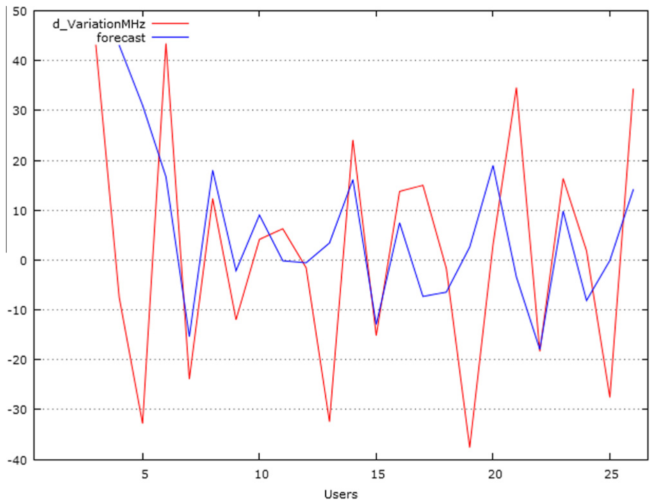

Fig. 10. Graph ARIMA model 1 Forecast prediction.

	d_VariationMHz	prediction
1		
2		
3	43.10	
4	-7.50	43.07
5	-32.80	31.05
6	43.44	16.68
7	-23.89	-15.37
8	12.33	18.01
9	-11.97	-2.15
10	4.14	9.02
11	6.27	-0.16
12	-1.56	-0.55
13	-32.44	3.41
14	24.08	16.13
15	-15.16	-12.93
16	13.76	7.47
17	14.99	-7.31
18	-1.70	-6.44
19	-37.62	2.60
20	3.10	18.97
21	34.59	-3.40
22	-18.31	-17.93
23	16.35	9.82
24	1.87	-8.13
25	-27.56	-0.09
26	34.28	14.11

Forecast evaluation statistics

Mean Error	-5.0957
Mean Squared Error	615.11
Root Mean Squared Error	24.801
Mean Absolute Error	18.066
Mean Percentage Error	68.041
Mean Absolute Percentage Error	6.3809
Theil's U	1.1117
Bias proportion, UM	0.042214

Fig. 11. Model I ARIMA (1,1,2) function evaluation (1).

The GNU software used to create the mathematical model is GRETL (GNU Regression Econometric and Time-Series Library) (GNU Force). This tool generates statistical analysis by means of correlograms and first and second derivatives to produce a stationary function in order to predict the behavior of a larger number of users vs. the CPU speed in MHz. These series present high frequency, no constant mean and variance, and multiple seasonality.

```
Function evaluations: 89
Evaluations of gradient: 30

Model 2: ARIMA, using observations 4-26 (T = 23)
Estimated using Kalman filter (exact ML)
Dependent variable: (1-L) d_VariationMHz
Standard errors based on Hessian

             coefficient   std. error      z        p-value
  ---------------------------------------------------------
  const        0.0890748   0.0929586     0.9582   0.3380
  phi_1       -0.246602    0.232310     -1.062    0.2885
  theta_1     -1.98185     0.243938     -8.124    4.50e-016 ***
  theta_2      0.999999    0.244297      4.093    4.25e-05  ***

Mean dependent var  -0.383478   S.D. dependent var   40.12179
Mean of innovations -7.602008   S.D. of innovations  15.24006
Log-likelihood      -100.2705   Akaike criterion     210.5411
Schwarz criterion    216.2185   Hannan-Quinn         211.9689

                       Real  Imaginary    Modulus  Frequency
  -----------------------------------------------------------
  AR
    Root  1          -4.0551     0.0000     4.0551     0.5000
  MA
    Root  1           0.9909    -0.1344     1.0000    -0.0215
    Root  2           0.9909     0.1344     1.0000     0.0215
  -----------------------------------------------------------
```

Fig. 12. Model I ARIMA (1,1,2) function evaluation (2).

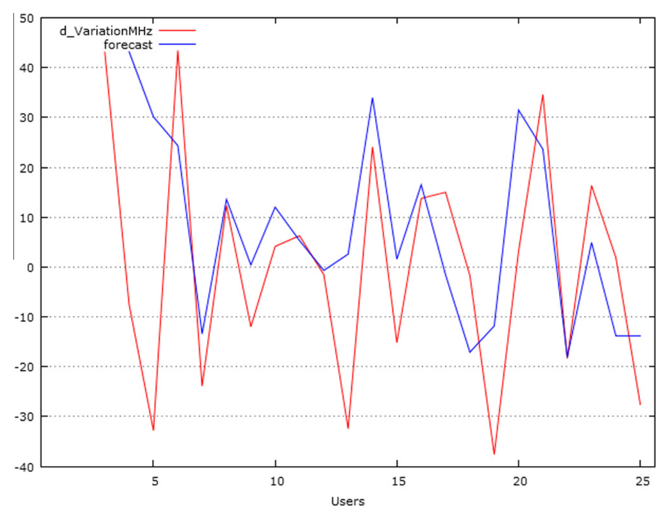

Fig. 13. Graph ARIMA model 2 Forecast prediction.

The Table 2 data, are represented in the following GRETL graphic (see Fig. 5):

After analyzing the statistical behavior is necessary to take the first difference as shown in the following figure (see Fig. 6):

This difference gives the graph of its Correlogram, which has 1 ACF point and 2PACF points (see Fig. 7).

To perform this observation, samples of 20 users crews were taken where 80% and 20% are left to perform the prediction (see Fig. 8):

According to the coefficients obtained from the autocorrelation functions and the partial autocorrelation prediction models, which were based on the coefficients p (Rho) AR(1) = −0,4498, $q1$ $q2$ MA(1) = −0,4498, and $q2$ MA(2) = −0,4079, the ARIMA model combination ARIMA(1,1,1) were defined with this equation:

$$(1 - 0,4498L)(1 - L)Y_t - (1 - 0,4498L - 0,4079)u_t \tag{2}$$

	d_VariationMHz	prediction
1		
2		
3	43.10	
4	-7.50	43.19
5	-32.80	30.13
6	43.44	24.35
7	-23.89	-13.40
8	12.33	13.57
9	-11.97	0.43
10	4.14	12.01
11	6.27	5.30
12	-1.56	-0.68
13	-32.44	2.59
14	24.08	33.97
15	-15.16	1.61
16	13.76	16.49
17	14.99	-1.53
18	-1.70	-17.08
19	-37.62	-11.83
20	3.10	31.49
21	34.59	23.62
22	-18.31	-18.09
23	16.35	4.91
24	1.87	-13.82
25	-27.56	-13.83

Forecast evaluation statistics

Mean Error	-8.5901
Mean Squared Error	522.86
Root Mean Squared Error	22.866
Mean Absolute Error	16.778
Mean Percentage Error	19.916
Mean Absolute Percentage Error	-1.229
Theil's U	0.84259
Bias proportion, UM	0.14113
Regression proportion, UR	0.1402
Disturbance proportion, UD	0.71868

Fig. 14. Function prediction calculated the RMSE (Root Mean Square Error).

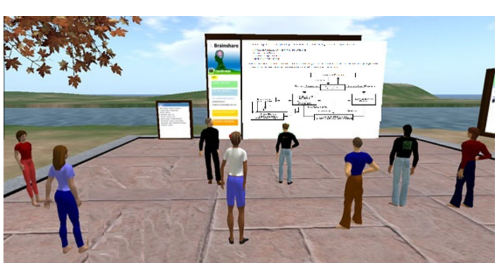

Fig. 15. Screen where there are several students with their avatars taking the course of expert systems using Sloodle presenter for the ARIMA model.

From the data provided in Fig. 9 the function prediction plot for 23 samples, the differentiated function is painted in red and the corresponding forecast as follows (see Fig. 10):

The Fig. 11 presents the exact data for the differentiated function and data for function prediction also calculated the RMSE (Root Mean Square Error):

From the data provided in Figs. 11 and 12 function prediction plot for 23 samples in red differentiated function is presented in red and the corresponding forecast (see Fig. 13):

The Fig. 14 presents the exact data for the differentiated function and the data for function prediction calculated the RMSE (Root Mean Square Error) too:

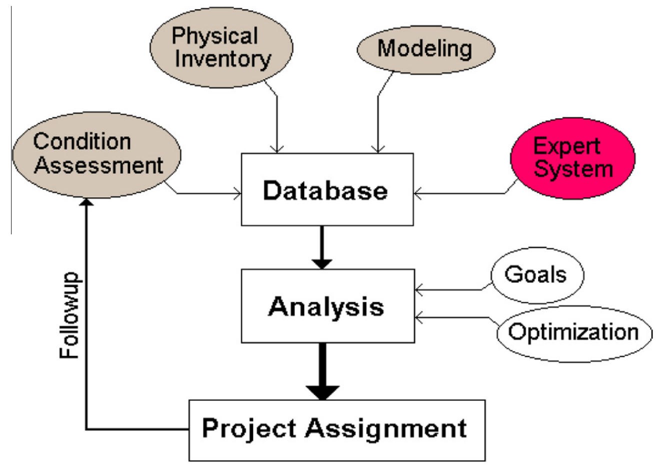

Fig. 16. Implementation of testing methodology for the ARIMA mathematical (GNU Force).

7. Results

The first step in creating an expert system course in a virtual world is to learn the different tools available for this purpose and how the same function can be properly used without generating errors at the moment of building these courses. The main difficulty found was to operate correctly, because virtual worlds had problems connecting to more than 25 viewers or students represented by their avatars (see Fig. 15).

There is also within the metaverse a course linked to the Moodle platform which is installed on the same Ubuntu server. In this platform test elements were created which were accessed from the course (see Fig. 16).

The ARIMA models in (1) and (2) have been applied to predict the behavior in terms of CPU server speed of the number of avatar access to the virtual 3D course, The Fig. 8 corresponds to the Autocorrelation function for ACF AR(1), PACF MA(1,2),

The first model ARIMA(1,1,1) for 23 samples shows mean errors around 5% (this prediction appears in Fig. 11), for the second model ARIMA(1,1,2) presents mean errors around 8%.

8. Conclusion

This paper analyzes the implementation of a metaverse targeted to educational virtual environments and the dissemination of scientific knowledge in the form of free open expert system courses. A simulation was carried out by means of GNU tools that control the characteristics of the objects created. It was motivated by the need experienced by different research groups of several Universities to study the utilization of free tools to set up science courses. In particular it has been shown how the application of OpenSim in specific areas of study in engineering has already made significant progress. A good example of application of these tools is the development of a expert system course with the aim of testing and proofing ARIMA mathematical models of behavior when the server has a high load of traffic generated by access of students who may wish to take the online courses. The ARIMA model produces an acceptable medium term forecast.

Acknowledgements

This proposal was carried out in the network of advanced technology research "RITA" of Distrital University Francisco José de Caldas of Bogotá, Colombia (Francisco José de Caldas Distrital

University, 1948), as part of the Doctoral work in Engineering, as part of the "LIDER" research group (Julian & Roberto, 2009).

References

Barbulescu, M., Marinescu, M., Grigoriu, O., Neculoiu, G., Sandulescu, V., & Halcu, I. (2011). GNU, GPL in studying programs from the systems engineering field. In *Roedunet international conference (RoEduNet)* (pp. 1–4). 10th, June 2011.

Brewster, C. O. (2004). Knowledge representation with ontologies: The present and future. *IEEE*, 72–81.

Buckley, J. J., & Eslami, E. (2002). *An introduction to fuzzy logic and fuzzy sets*. Berlin/New York: Springer-Verlag.

Cottrell, A., & Lucchetti, R. (2008). Gretl user's guide ARIMA time-series. *Università Politecnica delle Marche*, 139–146.

Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W., & Tuecke, S. (1997). A resource management architecture for metacomputing systems. In *IPPS/SPDP '98. Workshop on job scheduling strategies for parallel processing*.

Di Giacomo, M. (2005). MySQL: Lessons learned on a digital library. *Soft. IEEE, 22*(3), 10–13.

Dvorski, D. D. (2007). Installing, configuring, and developing with Xampp.

Erdmann, M., Fensel, D., Horrocks, I., Klein, M., & Melnik, S. (2008). The semantic web on the respective roles of XML and RDF. In *IEEE, in conference publication in the third workshop on workflows in support of large-scale science, on 17–17 Nov. 2008* (pp 1–10).

Fishwick, G. P. (2009). An introduction to Opensimulator and virtual environment agent-based M&S applications. In *Simulation conference (WSC), proceedings of the 2009 winter* (Vol. 64, pp. 177–183).

Francisco José de Caldas Distrital University (1948). Bogotá Colombia. Available from http://udistrital.edu.co.

Getchell, K., Miller, A., Nicoll, R., Sweetman, R., & Allison, C. (2010). Games methodologies and immersive environments for virtual fieldwork. *Learning technologies, IEEE transactions on, 3*(4), 281–293.

Giarratano, J. C., & Ricley, G. D. (1998). *Expert systems: Principles and programming* (third ed.). Florence, KY: Course Technology.

GNU Force, Development by the Gretl Team, Available from http://gretl.sourceforge.net/.

Gonzalez-Crespo, R., Rios-Aguilar, S., Ferro-Escobar, R., & Torres, N. (2012). Dynamic, ecological, accessible and 3D Virtual Worlds-based Libraries using OpenSim and Sloodle along with mobile location and NFC for checking. *International Jorunal of Interactive Multimedia and Artificial Intelligence, 1*(7), 62–69. http://dx.doi.org/10.9781/ijimai.2012.177.

Hannes, B., & Soren, A. (2005). Mapping XML to OWL Ontologies. Available at: http://www.informatik.unileipzig.de/~auer/publication/xml2owl.pdf.

Hippo OpenSim Viewer, Available at: http://mjmlabs.com/viewer.

Isaza, G., Castillo, A., López, M., & Castillo, L. (2009). Towards ontology-based intelligent model for intrusion detection and prevention. In *Computational intelligence in security for information systems. Advances in intelligent and soft computing* (Vol. 63, pp. 109–116). Springer.

Isaza, G., Castillo, A. G., López, M., Castillo, L. F., López, M. (2010). In *Proceeding of: information security and assurance – 4th International conference, ISA 2010*, Miyazaki, Japan, June 23–25, 2010. Proceedings 01/2010. pp. 51–63.

Julian, C., & Roberto F., (2009). Research Laboratory for Development Electronic and Networks. In *LIDER-UD*. Available from http://201.234.78.173:8080/gruplac/jsp/visualiza/visualizagr.jsp?nro=00000000005315.

Lala, D., & Nishida, T. (2011). Visie: A spatially immersive interaction environment using real-time human measurement. In *Granular computing (GrC), 2011 IEEE international conference on, Nov. 2011* (pp. 363–368).

Linden Research, Inc, Available at: http://lindenlab.com.

Pattal, M., Li, Y., & Zeng, J. 2009. 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).

Phankokkruad, M., & Woraratpanya, K. (2009). Web services for learning management systems: Communication architecture. In *Communications (MICC), 2009 IEEE 9th Malaysia international conference on, december* (pp. 403–408).

RealXtend Viewer, Available at: http://realxtend.org.

Russell, S., & Norvig, P. (2002). *Artificial intelligence: A modern approach* (second ed.). Upper Saddle River, NJ: Prentice Hall.

Simons, G. F., Lewis, W. D., Farrar, S. O., Langendoen, D. T., Fitzsimons, B., & Gonzalez, H. (2004). The semantics of markup: Mapping legacy markup schemas to a common semantics. In *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.

Spaeth, S., Stuermer, M., Haefliger, S., & Von Krogh, G. (2007). 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).

Tate, J., Bernasconi, A., Mescher, P., & Scholten F. (2003). Introduction to storage area networks. In *IBM red book*.

Torres, N., & Uribe, J., (2012). Design and analysis of connectivity platform based multicast IPv6 3D virtual environment for streaming and video repository on red RITA-UD QoS tests. Bogotá: District University.