# Prototyping 3D Virtual Learning Environments with X3D-based Content and Visualization Tools

*Livia Stefan*
National University of Arts, Bucharest, Romania
Calea Griviței 28, București, Tel. 021 314 3211
livia.stefan@yahoo.com

*Sorin Hermon*
Cyprus Institute, Nicossia, Cyprus
Tel. +357 22 208700
s.hermon@cyi.ac.cy,

*Marina Faka*
Cyprus Institute (CYI), Nicossia, Cyprus
Tel. +357 22 208700
m.faka@cyi.ac.cy

**Abstract**

Over the last decade, learning settings on SecondLife or OpenSimulator 3D online platforms are employed as a distance service by virtualizing educational spaces or even an entire university campus. To immerse and engage the learners in a 3D space, architectural and ambient objects need to be designed, besides the transfer of the educational assets in a 3D/2.5D environment. The present research addresses the existing limitations of the current 3D online platforms in terms of content creation, management and interoperability. Several open 3D and web-based technologies, e.g. X3D or WegGL, are investigated and evaluated for their possible use for complex 3D and non-3D content in online environments. A working methodology and recommendation of tools are proposed for a more productive development and a prototyping pipeline. The tools provide a higher level of interactivity and facilitate the content visualization in browsers and mobile devices.

**Keywords:** 3D Virtual Learning Environments, 3D modeling, X3D, prototyping

## 1. Introduction

Modern educational practice is using a blended paradigm in terms of mixing learning styles, technologies, and devices. In this context, the 3D online virtual worlds for learning (3DVLE) are employed as an available form of Virtual Reality (VR), consisting in an artificial and immersive environment for simulations, collaboration and social interactions, i.e. virtual classrooms, auditoriums or an entire university campus. One important feature of those environments is that users are creating the content, which can be retrieved from usage sessions. SecondLife (SL 2017) and OpenSimulator (OpenSim 2017) are the most common 3DVLEs with a high number of registered users, applications, and free available content. User immersion and engagement in an artificial environment is directly influenced by the graphical design of the 3D scenes and ambient details (Cudworth 2014).

The design of 3D learning settings based on complex architectural facilities highlights limitations in terms of content management and interoperability. For SL and OS-based 3DVLE, the limitations are mainly coming from the current client applications (viewers): low interactivity and a primitive content management.

Other VR-based applications, such as mobile Augmented Reality (MAR), are currently offering high-levels of technology-readiness for developing interactive and interoperable applications, due to their integration with open web and 3D standards, such as HTML5 and X3DOM, and standardization initiatives (Lechner 2010).

This research addresses the existing capabilities for content creation and prototyping for OpenSim-based 3DVLEs, taking as an example the creation of a 3D online environment representing campus educational buildings. The paper also investigates web-based standards and technologies, such as HTML5, X3D, WebGL, X3DOM and authoring tools, and presents results from several experiments and evaluations of different technologies, for their performances, openness, and standardization.

The X3D standard, meaning "extensible 3D graphics" (X3D 2017), originates from VRML (Virtual Reality Modeling Language) and employs a declarative programming model based on XML encoding for describing scenes and objects, visual effects and interactive 3D behaviors. It is maintained by the Web3D Consortium and it is ratified as an ISO standard (ISO/IEC 19775/19776/19777).

The development in 2009 by Fraunhofer Institute of X3DOM (X3DOM 2017), combining X3D, WebGL, HTML5, CSS, and JavaScript, eliminated the need for plugins and low-level programming. WebGL is used for rendering, while JavaScript interprets the DOM (Document Object Model) nodes of the X3D file and other associated XML documents.

Considering that standard browser-based technologies can be combined with the 3DVLE tools, the paper also proposes a working methodology and recommendation of tools for a more productive content development and prototyping pipeline, archiving of the digital assets, re-use and re-purpose. The tools provide a higher level of content interactivity and mobile client access.

Some parts of the experiments were performed at Cyprus Institute (CYI), the STARC department, and integrated in the doctoral research of the first author, regarding an experimental 3D virtual campus (Stefan 2015).

## 2. Research background and motivation

The 3DVLEs platforms are only "infrastructures" on which educational content must be built. Besides the 3D modeling of the environment, teachers have to design instructional materials and acquire specific 3D communication and interaction skills. The translation, even partially, of traditional courseware, objectives and pedagogical methods in a 3D dimension (or 2.5D) is a disruptive technological challenge. In other words, there is a technological barrier and a learning curve that teachers who wish to adopt 3D educational platforms have to overcome.

Many universities turn to commercial implementations (Daden 2017; DIGITALLY DESIGN 2017), but these are more focused on the technological aspects (Gütl 2011).

The current 3DVLE platforms are non-interoperable and evolve in parallel: Second Life, OpenSim, Project Wonderland (Fox 2010). There are many competitive graphic formats, but not one format accepted as a standard for content interoperability, and also no general methodology for creating virtual worlds (Zender et al. 2009).

The interactivity of the 3D content is provided by means of the viewer's editor and not as a feature of the virtual environment. The content is organized per user and per object inventories and does not feature a versioning system. The latest versions of the most important viewers, i.e. Singularity (SINGULARITY 2017) and Firestorm (FIRESTORM 2017) allow both the import and export of objects in Collada (Collaborative Design Activities) format (COLLADA 2017), which is an important step to use more complex 3D models in this kind of online mediums and to a greater interoperability among 3D platforms. Collada is a graphical object interchange XML-based description, with dae (digital asset exchange) extension for which third-party viewers are needed to visualize the 3D content.

The processing workflow for 3D modeling and design for a 3DVLE represents a time-consuming stage in the overall pipeline production. One reason is that a range of technologies and tools are typically used. In (Cudworth 2014) a 3-week period is indicated for experienced users to perform the 3D modeling of a virtual space. In our case, a 3-month work was needed for designing a working model of a 3D virtual campus (see Figure 1 and Figure 2 for final results).

From our first experiment, we identified the need for several models for different levels of details (LOD), content management and versioning, visual archives, access from mobile devices. Content prototyping and management are important factors for a complex 3DVLE comprising several architectural facilities and objects, in order to reduce the overall processing time for the 3D environment and facilitate the integration of the desired functionalities.

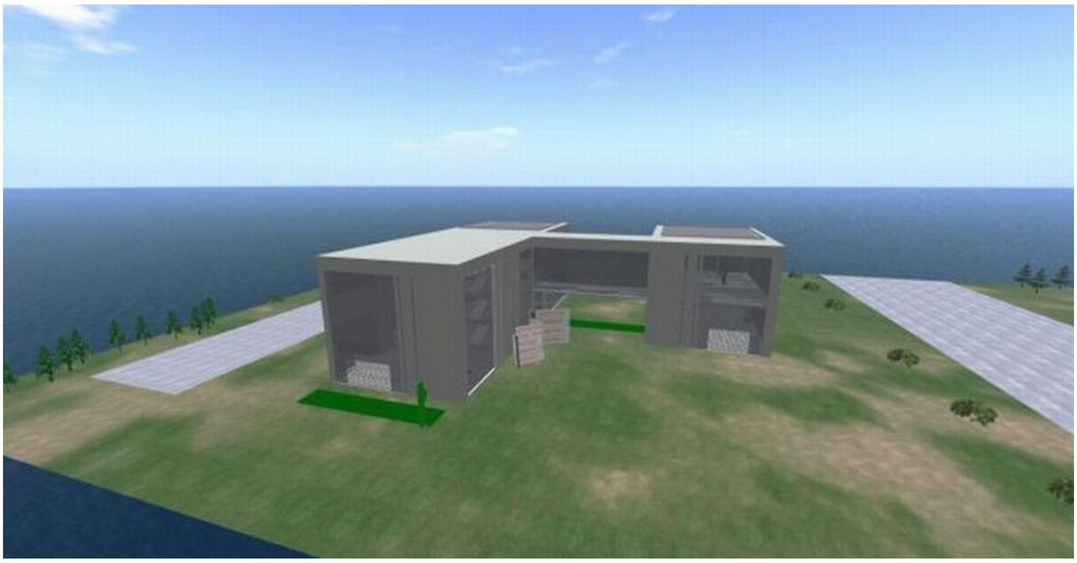

*Figure 1. The model of the 3D virtual campus - an outdoor view (3D modeling by Marius Hodea)*

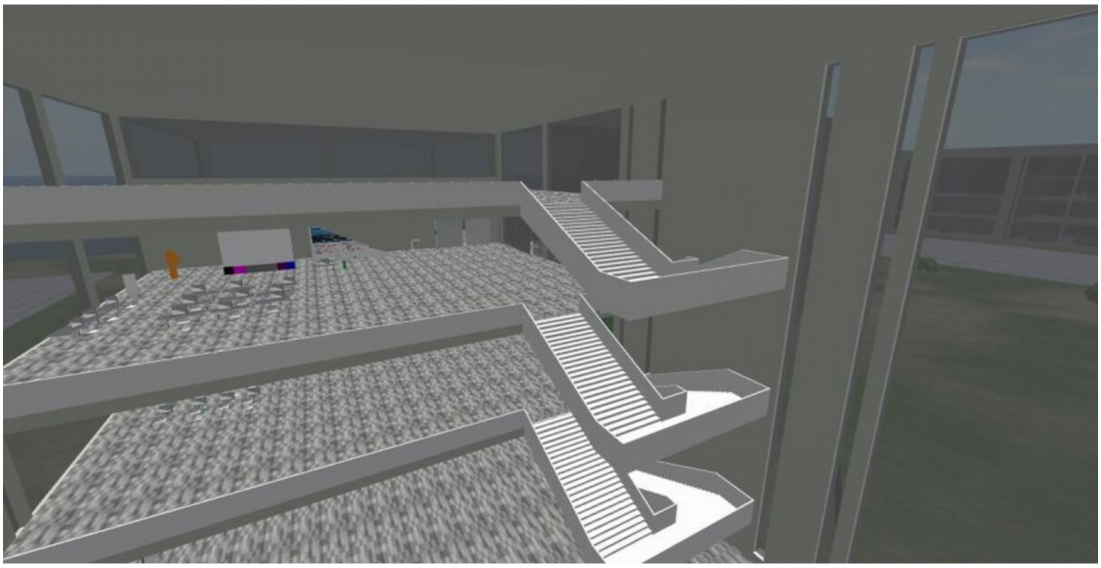

*Figure 2. The model of the 3D virtual campus - details from the building interior (3D modeling by Marius Hodea)*

**3. Lessons learned from similar research**

Content prototyping and management are typically applied in different domains, such as architecture, civil engineering or cultural heritage. We identified several research works which address the problem of content generation for 3DVLEs.

***3.1. Automatic generation of an educational space***

Freudenthaler (2011) provides a method for automatic instantiation of an educational space in Second Life from a web user interface. The method was applied for relatively simple educational spaces, which can be generalized to more complex environments.

Albion (2009) presents the Web3D project, a survey of open standards that enable viewing of 3D content in web pages. The research motivation was to simplify the access to 3D courseware for teachers who have no technical skills, in comparison with the tools offered by specialized 3D virtual environments (Albion 2009). The project offers examples of Web3D applications, development tools and a community of practice for teachers using Web3D. With these methods, the teachers involved in educational activities in 3D virtual worlds can focus on more important activities, such as the instructional design.

Quintella et al. (2010) examines the elements of virtual worlds that can be created using standards like X3D: a) 3D scenes; b) events in response to user interactions using ECMAScript scripting language and its extensions; c) data persistence through web services and ECMAScript that session state control; d) interaction with web interfaces; e) interaction of the users; f) integration with other applications.

The research results are several X3D tools and an application development toolkit (DWeb3D). For interaction with other applications, the X3D objects are converted into a Unity3D graph. For interaction with the web interface, Ajax3D was used. The paper concludes that X3D allows the development of virtual worlds as it can treat individual 3D worlds, but also presents several issues: X3D does not provide an easy environment for development and does not work well with big scenes. The usage of a high-level toolkit reduces the amount of code needed to create 3D applications.

Franco and Lopes (2013) propose an X3D-based modeling and programming learning model using a participatory approach of both teachers and students.

### *3.2. Prototyping the 3D content of an educational space*

Moro et al. (2010) present a method for automated building of a 3D virtual environment based on the analysis of images from a stereo camera. This method performs an automatic detection of regions of interest and extracts significant objects, which then are generated as 3D models. Statistical methods for image classification are used and also an algorithm for building 3D maps of objects that describe the shape, position and the approximate height of an object in the real environment. The method was tested and worked well for shaping an office with a desk and a chair. The method is an interesting research, but difficult to apply because it uses mathematical algorithms for training and classification that have not been tested on more complex graphical environments. The method opens an original perspective on prototyping content referring to interior design for virtual worlds.

Baldi & Lopes (2012) provide a set of 3D models for prototyping a complex campus infrastructure by using an OpenSim binary archive (OAR), which includes 3D content, scripts, and several personalized avatars. By installing the archive, users can make use of a functional, and also customizable virtual campus. The digital assets are not stored in viewer's inventory unless a specific action is performed on individually selected objects.

### *3.3. Expose of the 3D and non-3D digital assets*

FEDORA Commons (Flexible Extensible Digital Object Repository Architecture) is a "network or graph of interconnected digital objects" (Fedora 2017), launched officially in 2014. It allows the storage of "all types of content and associated metadata". FEDORA addresses organization's need of digital libraries and provides access services via RESTful APIs.

## 4. Critical considerations regarding OpenSim as an authoring platform

### *4.1. Several tools*

3D modeling is possible with the SL and OpenSim viewers' Build menu, which provide a pre-defined set of primitives, colors and textures, transformation tools (slice, twist, etc), and vegetation elements (grass, trees, bushes). The environment illumination and the particle effect system, e.g. snow, are also integrated into the viewers. Nevertheless, the viewer does not offer an effective tool for large models. Complex illumination techniques have to be implemented in more advanced 3D environments (e.g. employing the baking technique).

### *4.2. OAR Archives*

OpenSim-based 3DVLE can benefit from two types of archives, i.e. OAR, comprising the entire virtual world or certain regions with all their assets (3D objects, shapes, textures, terrain, scripts or notecards), and IAR, comprising only certain inventory elements. OAR can be used as a method for rapid generation of the virtual world, without objects being stored into the inventory, while, on the contrary, the IAR, stores the objects into the inventory, and lets the user make use of them to design a virtual world. In order to reuse and re-purpose the different assets, the owner of the objects must assign transfer and modify permissions. These permissions are not well recognized with different OpenSim versions.

### *4.3. Terrains*

The virtual terrain can be created or modified with 3D editing software (e.g. Photoshop or GIMP), redimensioned and divided among several regions.

### *4.4. Illumination*

The colors are affected by the ambient light from the virtual environment that can be solar or lunar. The ambient lighting can be eliminated with Full Bright option from the viewer's interface. A secondary effect is the elimination of the shadows. OpenSim does not have the capability to play specular reflections, but only to filter out the reflected light. These effects are resource consuming and can be eliminated with Bump making Enable / Shiny effects.

### *4.5. Level of details*

The management of different levels of details (LOD) in SL and OpenSim is performed manually in the viewer, by allowing the load of several models or of the original model with progressive modifiers (such as 0.5, 0.25, 0.125).

## 5. Web-related 3D open technologies and standards

A full integration with the web of the VR/AR applications is largely recognized as an important objective for richer applications and data integration from different data providers and devices.

### *5.1. X3D*

Integration of interactive 3D content and data exchange in web applications has been a challenge due to the limitations of the JavaScript language. With the advent of HTML5 and X3D, it is possible to run 3D interactive applications integrated in the web pages and therefore to interact by means of JavaScript. The roadmap started in 1995 with the development of the Virtual Reality Modeling Language (VRML) which supports 3D geometry, animation and scripting (De Luca et al. 2013) but the visualization depends on specific plugins, e.g. Cordona (CORDONA 2017), and it is decoupled from the HTML DOM (De Luca et al. 2013).

X3D as a superset of VRML is a mature ISO standard with relevant implementations, involving 3D content visualizations. From a programmatic point of view, X3D is a scene graph which supports, besides 3D graphics (polygonal geometry, parametric geometry, hierarchical transformations, lighting, materials and multi-pass, multi-stage texture mapping, animation) also humanoid animation and morphing; spatialized audio and video sources mapped onto geometry in the scene (Lee 2011); user interaction, collision, proximity and visibility detection; networking - ability to compose a single X3D scene out of assets located on a network; hyperlinking of objects to other scenes or assets located on the World Wide Web (through an anchor tag); user-defined objects –ability to extend built-in browser (Ranon 2015).

The X3D specification includes a web-browser integration model and the Scene Access Interface (SAI) which permits the manipulation of the X3D scenes both internally and externally. The web browser holds internally the X3D scene and an application that can handle the content using SAI. X3D comprises three encodings (ClassicVRML, XML, and Compressed Binary) and two SAI language bindings (JavaScript and Java) (WEB3D 2017).

The standard is already used in research projects regarding visualization of complex 3D content (De Luca et al. 2013; Jung, Behr & Graf 2011; Yung 2008; Zollner 2009).

### *5.2. X3DOM*

To solve the problem of a mechanism for updating or synchronizing 3D data not existing in SAI, Fraunhofer Institute, a Web3D Consortium member, in collaboration with W3C, has developed X3DOM, a JavaScript open-source framework (X3DOM 2017; WEB3D 2017; Behr et al. 2010). X3DOM performs a DOM integration for X3D and HTML5 and allows for a seamless integration of interactive 3D content into HTML pages based on a declarative programming model, similar to other standards such as Scalable Vector Graphics (SVG). The manipulation of the 3D content can be performed by adding, removing or changing DOM elements (Jung, Behr & Graf 2011).

The current release of X3DOM supports native implementations (iOS8, Chrome, and Firefox for Android), with fallback to WebGL API, and partially to X3D/SAI plugins (INSTANTREALITY 2017). X3DOM is above WebGL, OpenGL and DirectX, and subsequently has less complexity (in Figure 3 is shown the graphical stack). Integrated into the HTML DOM, X3DOM allows web programmers to continue their experience, based on known web technologies such as CSS, Java Script, JQuery or Ajax.

Standard technologies can streamline a VR or AR application development, by hiding the low-level complex tasks, and allow the access to device sensors and video camera via high-level API functions.

X3DOM supports embedded X3D-XML files references using inline nodes, i.e. an X3D-XML file can reference other X3D-XML files and build a hierarchy of assets (X3DOM 2017) which can be loaded in the background with a higher throughput.

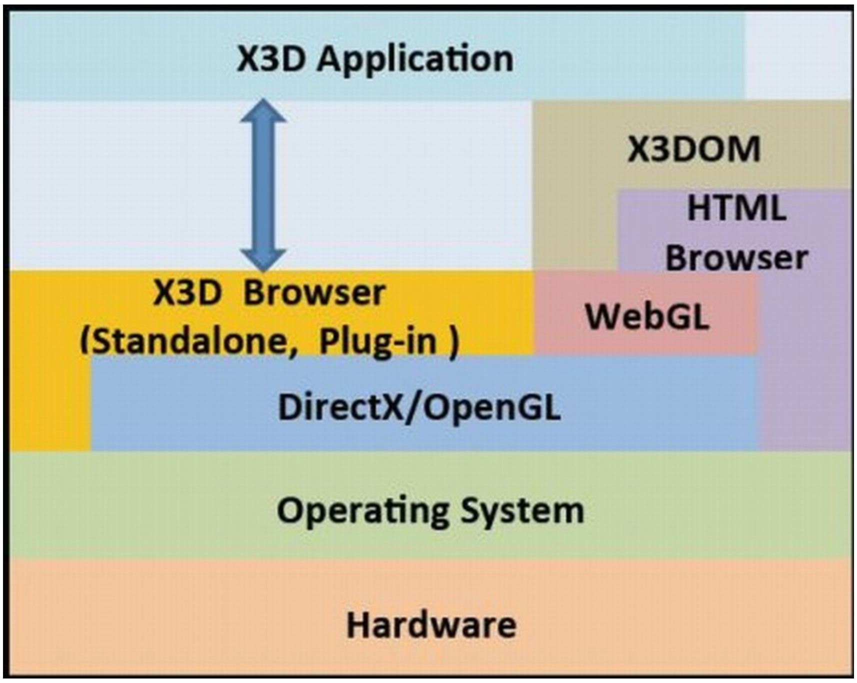


*Figure 3. The graphic stack of X3DOM (Havele 2011)*

### *5.3. WebGL*

A current implementation of X3D is based on WebGL (Web Graphics Library) (KRONOS 2017), which is a JavaScript API for rendering interactive 3D and 2D graphics within any compatible web browser and without the use of plugins. The Khronos organization promotes

WebGL as one solution for hardware accelerated 3D rendering in the web. WebGL is a cross-platform standard for a low-level 3D graphics API based on OpenGL ES 2.0, exposed through the HTML5 <canvas> element as Document Object Model interfaces. WebGL uses an imperative programming model.

## 6. Experiments and results

### *6.1. 3D modeling*

For our research, several iterations and methods were employed for the 3D design of an online campus. Different virtual models of a faculty building (see Figures 4,5) were designed and finally a virtual model of a 3D campus comprising a simplified 3-story faculty building (Figure 6) was created. The objective was the optimization of the 3D model and the demonstration of the desired functionalities. For these purposes two 3D modeling and post-processing software were used, i.e. 3DSMax and Trimble Sketchup. The model of the building resulted in 5962 vertices and 4528 faces. The textures and illumination were applied using OpenSim's in-world tools. Furniture objects (tables, chair, computer monitors) were taken from the Google 3D Warehouse, distributed and shared under Trimble General Model License.

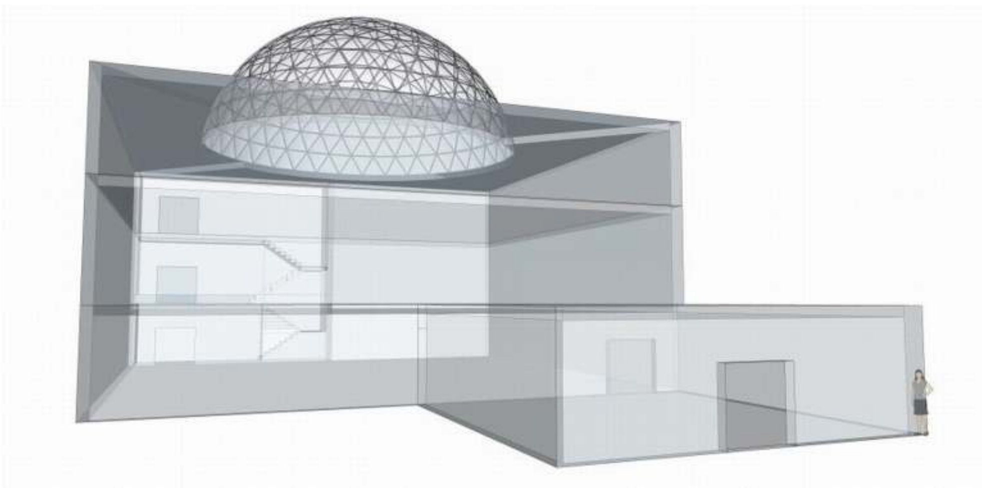

*Figure 4. Iteration of the 3D modeling of the virtual faculty building (3D modeling by Marius Hodea)*

*Figure 5. Iteration of the 3D modeling of the virtual faculty building (3D modeling by Marius Hodea)*

*Figure 6. The first implementation of the virtual campus (OpenSim import of the 3D model)*

### *6.2. X3D model in web page*

To integrate the model into a web page, a model conversion to X3D format and an X3DOM output under the form of an HTML5 encoded webpage (Figure 7) were needed. Instant Reality distribution provides a command line transcoding tool, named Avalon Optimizer (aopt), that was used to convert a VRML format (wrl extension) of the model to X3D. The *aopt* utility also performed a mesh restructuring to increase the rendering performance, using the following command (AOPT 2017):

```
aopt -i [input.foo] -F "Scene:opt(1),maxtris(20000)" -x [output].x3d
```

```
<x3d  id='x3dElement'  showStat='false'  showLog='false'  style='width:100%;  height:100%;
border:0; margin:0; padding:0;'>
      <scene DEF='scene'>
        <group>
          <matrixTransform matrix='1 0 0 0 0 0 -1 0 0 1 0 0 0 0 0 1'>
            <group>
              <matrixTransform matrix='1 0 0 0 0 1 0 0 0 0 1 0 979.726 583.542 0 1'>
                <shape>
                  <appearance DEF='ID4'>
                    <material       diffuseColor='0.886274       0.886274       0.886274'
shininess='0.078125' specularColor='0.4 0.4 0.4'></material>
                  </appearance>
<indexedTriangleSet creaseAngle='1.5708' solid='false' index='0 1 2 1 0 3 4 5 6 7 6 5 8 9
10 9 8 11 12 13 14 15 14 13 16 17 18 17 16 19 20 21 22 23 22 21 24 25 26 25 24 27 28 29
30 31 30 29'>
<coordinate point=''></coordinate>
              <normal vector=''></normal>
</indexedTriangleSet>
                </shape>
              </matrixTransform>
            </group>
          </matrixTransform>
        </group>
      </scene>
</x3d>
```

*Figure 7. The HTML source (partial) code, integrating the X3D model*

The pipeline processing was the following: a) the 3D model from Sketchup was saved as a Collada file; b) this file has been imported in MeshLab (MESHLAB 2017) and converted to VRML97 format (wrl); c) aopt utility was used to convert wrl files to X3D and HTML5 files.

The model was visualized in the OpenSim virtual world setting using an external browser (Figure 8).

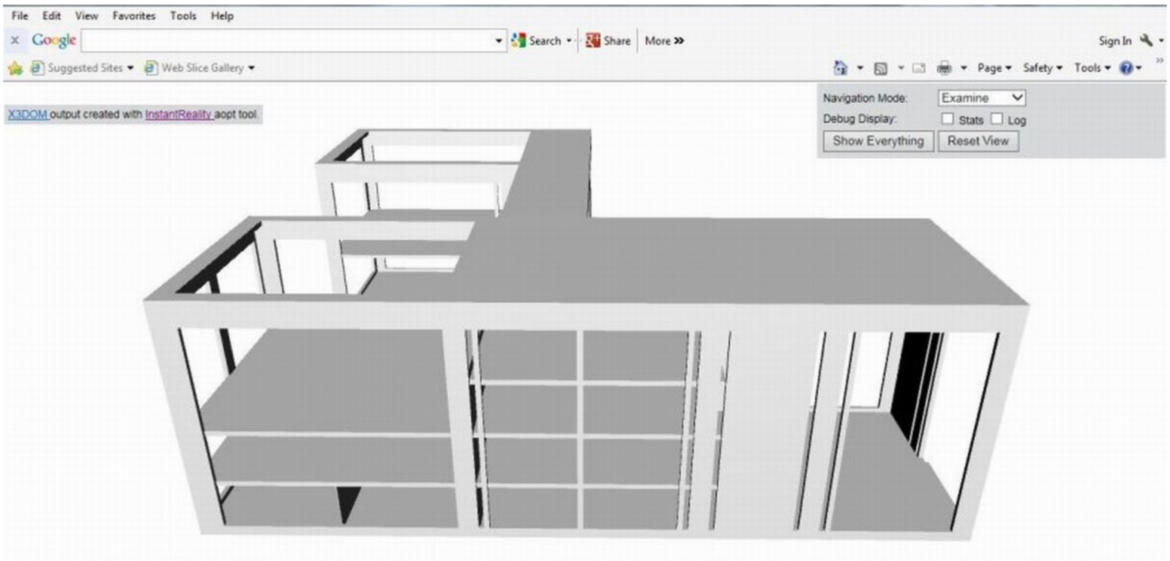


*Figure 8. The 3D model of the faculty building in a HTML page*

### *6.3. Laser scanning*

Laser scanning uses a light source that is projected on the surface of the object and an optical detector to extract the shape of the scanned object. This process usually uses a triangulation principle.

For our experiment, a Laser Scan NextGen have been used along with post-processing in MeshLab to digitize an ambient object. The resulted mesh contained 10 layers, which were flattened on different groups to fill the goals and to create a consolidated composition. For the texture, several photos were applied and then aligned (registered) with the mesh. The mesh model used the Poisson algorithm and was saved using the Standard Stanford Polygon Format *(*PLY) and the VRML97 formats, to be prepared for X3D/X3DOM conversion and encoding.

### *6.4. Photogrammetry and structure from motion*

Structure from motion is a photogrammetry technique, i.e. the generation of 3D digital content from successive and overlapped photos. It is a method for rapid creation and post-processing of a 3D model, with lower precision than the previous method. If professional DSLR cameras are used, the precision can be improved.

For our experiments, Autodesk 123D Catch21 was used. This application has also a mobile version for IOS and Android that allows the use of smartphones for taking pictures in an outdoor context. The images are stored on a local computer or uploaded to Autodesk 123 Catch cloud system to generate the 3D models. Sketchfab (SKETCHFAB, 2017) or AgiSoft (AGISOFT 2017) can also be used as commercial software packages.

Structure from motion was used for rapid modeling of a small room with all its objects. Two files were generated, a Wavefront obj and mtl (corresponding to the texture). The 3D model was post-processed with MeshLab, during which several filters were applied to clean up the model. The mesh model was also connected with the scanned model, by choosing at least 4 connection points. The 2D and 3D results are shown in Figures 9, 10. A post-processing could also be performed using the Autodesk 123D Catch web application.

The model was further converted in VRML97 to be prepared for X3D/X3DOM conversion and encoding. The model was also visualized in OpenSim virtual world using an external browser.

*Figure 9. Room model generated with Autodesk 123D Catch - the 2D model*

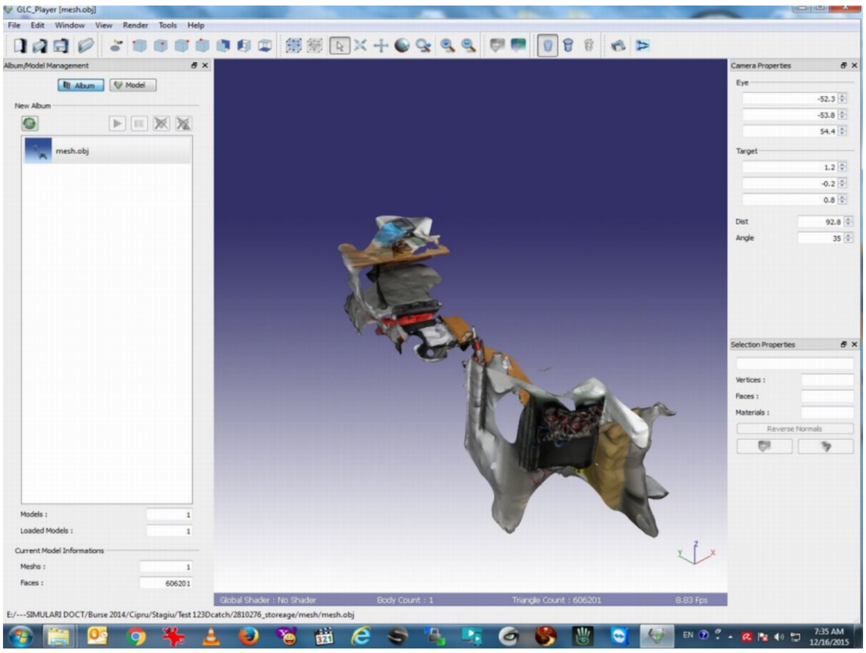


*Figure 10. Room model generated with Autodesk 123D Catch - the 3D model (screen capture from GLC Player)*

**7. Discussion**

The 3D model usually represents the highest level of detail (LOD). Several important constraints need to be applied to 3D mesh models for SL and OpenSim platforms: texture dimension has to be a power of 2 and lower than 1024 x 1024; combining textures in a single UV map (UV map); a maximum of 8 materials; a maximum size of 64x64x64m; saved as Editable Poly with Transforms Reset, and stack collapse.

As SL and OpenSim are real-time rendering systems, the 3D models have to be optimized. On one hand, there is a need to create realistic environments, and on the other hand, the 3D content needs to be optimized for a real-time and distributed environment (e.g. the avatar is moving from one place to another and the scene has to be updated on several concurrent clients). The mesh optimization for import in SL and OpenSim refers to: a) models with fewer polygons; b) elimination of double-sided faces; c) back-face culling, conversion of curves and surfaces; d) face triangulation (BLENDER 2014). The benefits of using 3D meshes for complex objects such as buildings are resulting in higher levels of realism. Optimized mesh objects not only improve the graphics, but also the performance, as they put a less server load and facilitate an improved client rendering. Meshes also offer superior physics (e.g. avatar/object collisions).

With the use of 3D laser scanners, real objects with complex characteristics can be digitized with a higher accuracy. This method resulted in a 3D data under the form of large unstructured point clouds which were post-processed with the MeshLab application. For better results, the objects have to be colored and not shiny, not to absorb or reflect the light.

Using structure from motion-based photogrammetry the most important factor is the quality of the camera. No zoom has to be applied and the camera motion has to be circular and steady.

The structure from motion has the advantage that 3D models of complex objects can be obtained, with minimal post-processing effort. The objects can also be an environmental object, such as a tree. Even with this method, large objects are generated, in our case, a model of 56 Mb. With smartphones, different objects from outdoor settings can be acquired, streamlining the process of digitization of real models.

## 8. A proposed methodology

For 3D modeling of buildings and interior spaces of a complex 3D online architectural environment, we recommend Trimble Sketchup. The spaces have to be designed according to the contemporary interior design principles. It is important to pay attention to lighting, space organization and ergonomics of furniture and utilitarian objects, to create a psychological comfort and to engage the users inside the virtual space. It is also necessary an “inclusive design” to accommodate special needs (Cudworth 2014; DESIGN FOR ALL 2017).

To perform a rapid prototyping, our approach is closer to that described in Baldi & Lopes (2012). The graphical environment (the “simulator”) and the functional scripts can be rapidly reproduced by reinstalling an OAR archive on other OpenSim instances. OpenSim archives provide an efficient prototyping method for an entire simulator. Therefore, it is a viable method for the transfer and re-use of an infrastructure describing a complex virtual world, with all its elements, 3D content, and scripts. To make use of the extensibility features of OpenSim platform, the new simulator can be attached to other existing simulators. Using the OpenSim viewer’s tools, the management of further modifications of the content is not productive. These actions can be performed by storing assets in viewer’s inventory or by exporting objects in Collada format.

Instead, the versioning and management can easily be performed by means of metadata, a capability of the X3D format. The downside of this method is the need to learn a new format, which can be overcome by employing the aopt utility (AOPT 2017) and the existing JavaScript frameworks, e.g. WebGL.

Prototyping can also include the development of toolkits for automatic content generation simulator, but in the case of an architectural environment, the components are too complex to be automatically generated. Furniture elements or the learning artifacts (i.e. content created by learners) can be converted to be viewed in X3D compatible browsers or included in online galleries (Figure 11). After functional and 3D content prototyping, certain components of the virtual campus can be easily modified and adapted as needed.

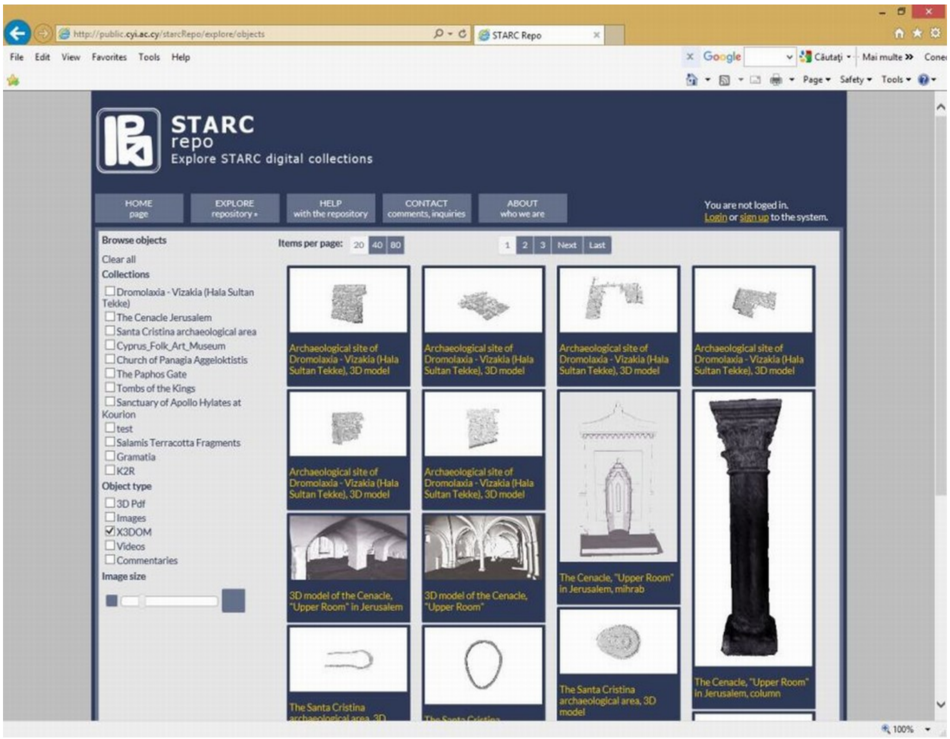


*Figure 11. Online accessible repository of digital data on cultural heritage with X3D models (STARC Web Repository, 2017, © Copyright 2017, STARC, Cyprus Institute. Used with permission)*

The chart below (Figure 12) summarizes the workflow recommended for the implementation of a prototype of a 3D online campus.

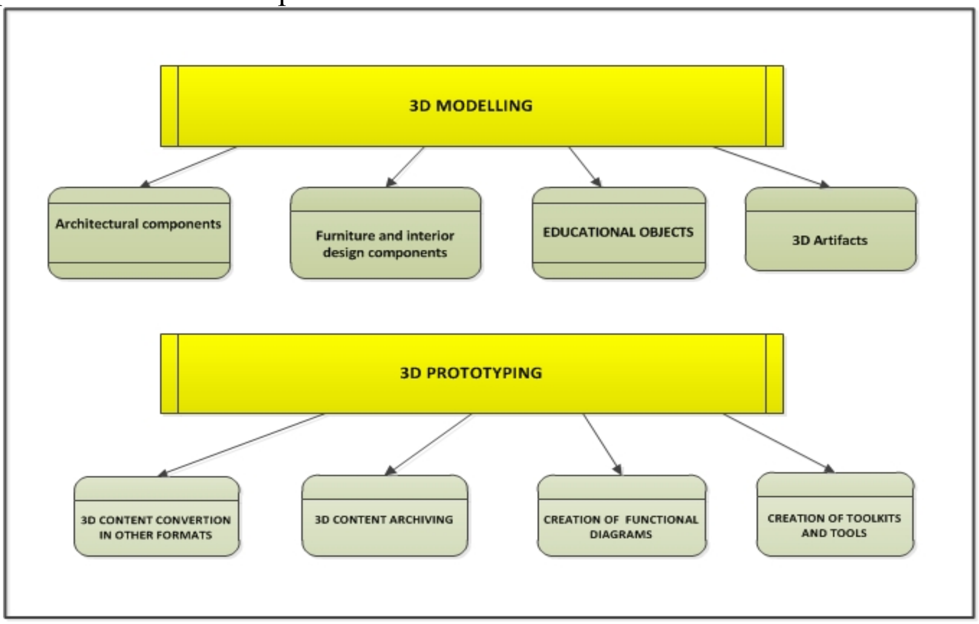


*Figure 12. Diagram of the proposed working methodology*

## 9. Conclusions and future work

In this paper, we reviewed alternative methods for production and management of complex 3D content, based on a critical analysis of the OpenSim platform and also on several experimentations of a 3D virtual architectural environment for educational purposes.

The use of X3D related web standards for describing 3D and non-3D content for online virtual worlds is recommended due to their capability to facilitate the prototyping, management and visualization with different visualization patterns, and also to provide an increased content interactivity. These standards also facilitate the creation of platform agnostic galleries of models, cross-platform integration, including mobile devices.

The X3D-based technologies are also important development tools for mixed realities, e.g. integration of 3D scenes and objects with real-time video, or devices and sensors.

The integration of 3DVLE with future equipment which will replace the traditional user interfaces, facilitated also by special X3D nodes, represents a challenge of further research work.

**Acknowledgments**

The research work of the first author was supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and the Romanian Government under the contract number POSDRU/159/1.5/S/137390/, and took place at University "Politehnica" Bucharest, NUA, and CYI. Authors thank Professor Dragoş Gheorghiu, NUA, for his architectural guidelines, PhD student Marius Hodea, NUA, for 3D modeling the campus building, and Bogdan Căpruciu for proofreading the text.

**References**

Albion, P. (2009). Mere mortals creating worlds: low threshold 3D virtual environments for learning. The International Journal of Learning, 16 (6). Retrieved from http://eprints.usq.edu.au/5803/2/Albion IJL 2009 PV.pdf.

Baldi, P., & Lopes, C. (2012). Universal Campus. An open virtual 3-D world infrastructure for research and education, ACM Library, DOI: 10.1145/2181207.2206888.

Behr J., Dähne P., & Roth M. (2004). Utilizing X3D for Immersive Environments, Proceedings of the 9th International Conference on 3D Web Technology, California, 71-78.

Behr, J., Jung, Y., Keil, J., Drevensek, T., Zöllner, M., Eschler, P., & Fellner, D. W. (2010). A Scalable Architecture for the HTML5 / X3D Integration Model X3DOM. 15th International Conference on 3D Web Technology. New York : ACM Press, 185-193.

Cudworth, A. (2014). Design of Virtual Worlds: Creating Immersive Virtual Spaces, New York: Taylor & Francis, ISBN: 9781466579613.

De Luca, D., Guidazzoli, A., Fanini, B., Baglivo, A., Delli Ponti, F., & Liguori, M. C. (2013). In Addison, A., De Luca, L., Guidi,G., Pescarin, S. (Eds.), Proceeding of 2013 Digital Heritage International Congress, Marseille, France, IEEE, Vol. II, 711 – 718.

De Lucia, A., Francese, R., Passero, I., & Tortora, G. (2009). Development and evaluation of a virtual campus on Second Life: The case of SecondDMI. Retrieved from https://dl.acm.org/citation.cfm?id=1465378.

Fox, M., R., Kelly, H., & Patil, S. (2010). Medulla: A cyberinfrastructure-enabled framework for research, teaching, and learning with virtual worlds. In W. S. Bainbridge (Ed.), *Online worlds: Convergence of the real and the virtual* (pp. 87-100*).* London: Springer. http://dx.doi.org/10.1007/978-1-84882-825-4 7.

Franco, J. F., & Lopes, R. D. (2013). Using X3DOM for promoting students' digital and visual literacy at k-12 education: a use case, Proceedings of the 18th International Conference on 3D Web Technology, ACM, 213-213.

Freudenthaler, S. (2011). Flexible learning settings in Second Life. Unpublished master's thesis. Retrieved from http://www.iicm.edu:8080/home/cguetl/pivate/MAExamples/MasterThesis_StefanFreudenthaler_20101229_FinalVersion.pdf.

Gütl, C. (2011). The support of the 3D virtual worlds for enhancing collaboration in learning settings. In D. P. Francesca Pozzi (Ed.), *Techniques for fostering collaboration in online learning communities: Theoretical and practical perspectives* (pp. 278-299). Hershey: IGI Global.

Havele, A. (2011). Augmented Reality Roadmap for X3D. Position paper. 4th International AR Standards Meeting, Web3D Consortium, Basel, Switzerland. Retrieved from http://www.perey.com/ARStandards/[Haevle]Web3D_AR_Roadmap.pdf.

Jung, Y. (2008). Building Blocks for Virtual Learning Environments. WSCG 2008 Communication Papers. The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision in co-operation with EUROGRAPHICS, 137-144.

Jung, Y., Behr, J., & Graf, H. (2011). X3DOM as a carrier of the virtual heritage. Proceedings of 4th ISPRS International Workshop 3D-ARCH 2011, 475-482.

Lechner, M. (2010). A Proposed Step-By-Step Guide to an AR Standard, Wikitude, International AR Standards Group site. Retrieved from http://www.perey.com/ARStandards.

Lee, G. (2011). Supporting Mixed & Augmented Reality Visualization in X3D. AR/MR Workshop, Web3D, IT Lab NZ. Retrieved from http://web3d.org/membership/login/groups/Korea%20Web3D%20Forum/20090707-MR-Web3D-eng.pdf.

Moro, A., Mumolo, E., & Nolich, M. (2010). Building virtual worlds by 3D object mapping, Proceedings of the 2010 ACM workshop on surreal media and virtual cloning, 31-36.

Quintella, F., Soares, L. P., & Raposo, A. B. (2010). DWeb3d: a toolkit for developing X3D applications in a simplified environment. Proceedings of the 15th International Conference on Web 3D Technology Web3d'10, 45-54.

Ranon., R. (2009). X3D Introduction, University of Udine, Italy, Retrieved from http://hcilab.uniud.it.

Ştefan, L. (2015). Mixed-Reality Adaptive 3D Multi-User Online Communities Of Practice In Academic Education Tackling Students Motivation And Teachers' Self-Efficacy, Doctoral Consortium of the International Conference on Computer Supported Education CSEDU (DCCSEDU 2015), 16-22.

Zender, R., Dressler, E., Lucke, U., & Tavangarian, D. (2009). Pervasive media and messaging services for immersive learning experiences. Pervasive computing and communications, Percom 2009, IEEE International Conference, 1-6.

Zollner, Z. et. al. (2009). An Augmented Reality Presentation System for Remote Cultural Heritage Sites. 10th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST 2009, 112-116.

AGISOFT, www.agisoft.com, Accessed 2017.

AOPT, http://www.x3dom.org/documentation/tutorials/generic-3d-data-conversion, Accessed 2017.

AUTODESK 123D Catch, http://www.123dapp.com/catch, Accessed 2017.

BLENDER, www.free-lsl-scripts.com/Secondlife/Posts/Blender-Tips, Accessed 2017.

COLLADA, www.khronos.org/collada, Accessed 2017.

DADEN, www.daden.co.uk, Accessed 2017.

DESIGN DIGITALLY, http://www.designingdigitally.com, Accessed 2017.

DESIGN FOR ALL, www.designforall.org, Accessed 2017.

FIRESTORM, http://www.firestormviewer.org, Accessed 2017.

KRONOS, www.khronos.org/webgl, Accessed 2017.

INSTANTREALITY, www.instantreality.org, Accessed 2017.

MESHLAB, www.3d-coform.eu/index.php/tools/meshlab, Accessed 2014.

SECONDLIFE, http://secondlife.com, Accessed 2017.

SKETCHFAB, https://sketchfab.com, Accessed 2017.

SINGULARITY, http://www.singularityviewer.org, Accessed 2017.

STARC WEB REPOSITORY, http://public.cyi.ac.cy/starcRepo/explore/objects/40, Accessed 2017.

OPENSIMULATOR, http://opensimulator.org, Accessed 2017.

X3D, http://www.web3d.org/x3d/what-x3d, Accessed 2017.

X3DOM, www.x3dom.org, Accessed 2017.

WEB3D, www.web3d.org, Accessed 2017.

**Livia ŞTEFAN** received her MSc in Telephony and Data Transmissions from the Faculty of Electronics and Telecommunications, University POLITEHNICA Bucharest (1987). In September 2014 she performed a research stage at the Cyprus Institute (CYI). In September 2015 she received her Ph.D. degree at the University POLITEHNICA of Bucharest, concerning e-learning in 3D online virtual worlds. Currently, she works as a researcher and senior software engineer at Vauban IT Bucharest and collaborates with National University of Arts (NUA) in research projects. Her current research interests include e-learning, virtual and augmented reality environments, big data in education. She is a professional member of IEEE Computer Society and Association for Computing Machinery (ACM).

**Sorin HERMON**, currently Associate Professor in Digital Cultural Heritage at CYL, responsible for academic and research activities in the field of digital cultural heritage. He received his Ph.D. in 2002 and performed post-doctorate studies in scientific visualization for cultural heritage at PIN, University of Florence. Sorin Hermon is leading several EC funded research activities focusing on 3D scientific visualization for research and communication of Cultural Heritage, ontology and semantic data representation and virtual museums.

**Marina FAKA**, joined CYL in 2012 as a researcher assistant and participated in several projects considering the 3D documentation of archaeological sites/monuments. Her current position is technical research specialist as a geodesic engineer. She holds a 5-year engineering diploma from the School of the Rural and Surveying Engineering at the National Technical University of Athens (2006-2011).