

Copyright © 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Configurable and Flexible Immersive Learning Environment

An Enhanced Solution for the OpenSim Platform to Support End-users

Christian Guetl, Karl Haas

Curtin University of Technology, Perth, WA.
Graz University of Technology,
Graz, Austria
cguetl@icm.edu; karl.haas@alumni.tugraz.at

Vanessa Chang

Curtin University of Technology,
Perth, WA.
vanessa.chang@curtin.edu.au

Abstract — Information and communication technologies have influenced our way not only to acquire and share knowledge but also to collaborate and communicate. Consequently, this has strongly influenced the way how communities or learners share knowledge and collaborate in various learning environments and settings. In particular learning and collaborating in immersive environments are proving to be effective in areas such as skills training. However, early development of immersive learning settings follow the traditional methodology of designing and developing environments for a specific learning goal or activity. The literature review of previous experiences revealed that new forms of learning require flexibility and adaptation towards the learning groups' needs. This situation has motivated our international research collaboration to focus on approaches for adaptable configuration with reuse of learning settings in immersive environments. A promising proof of concept for the Second Life platform has led to an improved approach for the open source platform, OpenSimulator. This paper reports about the conceptual approach, design and development as well as findings for further work. The main contribution in the paper covers a Web-based configuration tool. It supports end-users (learners and teachers) to design and reuse learning settings that include 3D components and in-world tools. Managed settings are automatically created in in-world instances of OpenSimulator which caters for on demand requests.

Keywords- immersive learning environment; OpenSim; flexible learning settings; configuration; virtual world

I. INTRODUCTION

The world has changed dramatically over the last few decades. The fast progress and development in technology has increased the demand of skilled and educated members of our modern society. The impact of the rapid development in information technology and communication (ICT) has also affected the education front. Learning has become more self-directed and self-reflective resulting in the need to develop independent learning activities which are responsive to the needs of learners (Gütl & Chang, 2008).

Web-based tools for communication, active participation and collaboration (Daniel, 2012; Gütl, 2008) are powerful when used effectively to support learning and teaching

Today's digital immigrants and digital natives or Generation Y are experienced users of technologies. The day-to-day activities of today's knowledge learners and knowledge workers included the use of the web and social media tools to inform, learn, work and interact with one another. The concepts of 'learning anytime and anywhere' and 'learning on demand' are become more realistic. As a result, the need for more flexibility in learning settings has been identified (Chang & Gütl, 2010).

Over the last decade immersive 3D environments have become increasingly popular for various forms of communication and collaboration including educational settings (de Freitas, 2008). Immersive learning environments such as serious games applications and virtual world platforms provide a sense of presence, awareness of group activities, increased motivation and suppression in feeling isolation (Gütl, 2011). Immersive learning environments can support a broad spectrum of learning settings of distance education, such as collaborative learning in geographically dispersed groups (Chang et al., 2009), and in blended learning approaches, such as simulations or skills training in contextualized environments (Reiners et al., 2012). In recent years, a great number of virtual world learning environments have been researched and developed, (de Freitas, 2008). Such immersive learning settings follow traditional approaches in designing and developing environments for specific learning goals or activities. Usually resources and effort are invested heavily in these immersive learning environments which tend to be inflexible and highly tailored for specific educational goals and scenarios. However, practical experiences and literature reveal that flexibility is required in terms of how people learn and how to support them in the learning process with specific learning tools (Chang & Gütl, 2010). However, such environments are used only occasionally or for several hours per day with 24/7 deployment of information technology (IT) support resulting in wasted resources. Thus, creation of specific learning environments on demand would be beneficial.

This situation has motivated our international research collaboration between Curtin University in Western Australia and Graz University of Technology in Austria to focus on approaches for easy configuration, adaptation and

reuse of learning settings or learning spaces in immersive environments. The main idea of our proposed approach is to provide for end-users (teachers and students) an easy to use platform to manage their learning environments. 3D objects or inventory (such as tables and chairs) as well as collaboration and learning tools (such as whiteboards and presentation walls) can be selected and placed in the learning environment and the look (or set-up) of the learning room or space can be configured. Additionally, learning spaces or learning settings can be booked for a defined schedule and will be automatically instantiated in-world for a particular learning group.

An extensive literature survey has revealed that there is not much related or similar work. *Sloodle* is an open software that links the LMS Moodle with Second Life, enables data exchange and some learning tools to be added in-world from the avatar's inventory (Leidl & Rößling, 2007). *StellarSim* supports the automatic creation of 3D objects for simulation in astrophysics domain and has been included as plug-in architecture for OpenSim (Henckel & Lopes, 2010). The DWeb3D Toolkit aims to provide a set of tools to reduce the effort of developing 3D applications with X3D (Quintella, et al., 2010). The *automatic construction of virtual world environments*, irrespective of objects, is performed by means of a stereo image of the real world scenario (Moro et al., 2010). Virtual world platform *Open Wonderland* provides a *module concept* where complex 3D objects and tools can be installed and easily used and placed in-world (Kaplan & Yankelovich, 2011). Related research, developments and tools provide users or developers to simplify the process of learning environment creation in-world, for details refer also to (Gütl et al., 2010; Haas, 2012). To date and based on our review, there is no approach which supports the design, reuse and adaptation of learning settings by a simple user interface out of world and automatically creates the learning settings on demand in-world.

A first proof of concept has incorporated a Web-based application enabling end-users to design, reuse and adapt learning settings by a simple 2D drag and drop interface. An intelligent room manager object in Second Life (scripted object) communicates via a control interface with the Web application and automatically creates and removes learning settings. Although the concept has been proven feasible, the first solution has shown a number of issues and limitations; for details refer to Gütl et al., 2010. In this research we will report on an enhanced approach supporting different virtual world platforms and the development details for the OpenSimulator platform. For more technical details refer to Haas (2012).

This paper is outlined as follows: Section I details the literature review and motivation for the research; Section II outlines the requirements and the conceptual architecture for an intelligent learning room and space management. Section III gives details of the development of the Room or Space Manager application and the support for the OpenSim platform as well as discusses findings from the development work. Section IV concludes the research and gives some possible directions for further research.

II. REQUIREMENTS AND DESIGN

The aim of the research to develop an intelligent learning space that is adaptable and easily configurable to accommodate learning activities in an on-demand in-world learning setting. As such, the requirements are (1) a simple tool for end-users to create, manage, reuse and adapt learning settings with the use of pre-existing 3D objects and learning tools; and (2) efficient utilization of hardware and IT resources. Charged with the above, the development on a conceptual level incorporated the following requisites:

- Support different 3D virtual world platforms
- Enable an easy way to define, manage, reuse and adapt pre-existing settings for end-users (teacher, learner, instructional designer)
- Design the learning settings with the use of pre-existing 3D objects as well as the use of learning and communication tools
- Create the learning settings automatically in-world for a predefined time for specific use
- Support room or space management to schedule the provision of specific learning settings and handle access rights for learning groups
- Store a repository of the learning settings states or configuration at the end of the learning session to enable a recall of a state for future activities.

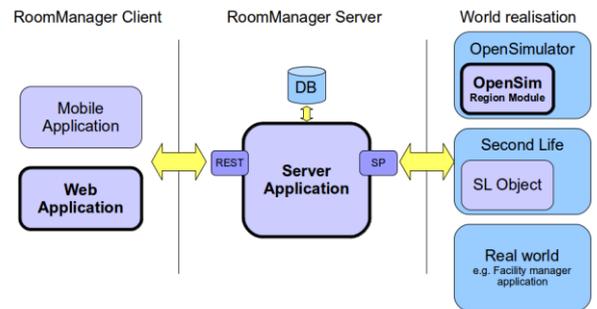


Figure 1. Generalized conceptual architecture supporting different client solutions and virtual world platforms (Haas, 2012)

Based on the requirements stated above and our previous experiences (refer to Gütl et al., 2010), the design follows the conceptual architecture depicted in Figure 1. For highest flexibility and extensibility, the conceptual architecture is built of three parts: the *RoomManager Server* is the core element which manages the learning settings and related information as well as provides interfaces for communication with the other systems. The *RoomManager Client* provides the graphical user interface and enables the interaction by the end-users; communication with the server makes use of REST (representational state transfer) architectural style. The *World realization or World Instances* refer to different 3D virtual world platforms, and in a generalized view, also to real world learning room and space settings, the *RoomManager Server* can interact by means of a server socket port (SP). The conceptual design supports various

user client applications and 3D virtual world platforms, however, the further scope of this work and implementation details are restricted to the bold rendered conceptual components of Figure 1.

III. DEVELOPMENT ASPECTS OF THE PROTOTYPE

A. Room Manager Server and Client

Following the conceptual design, the conceptual components of the room management have been separated from the virtual world environments and split into a server component and a client application.

The *RoomManager Server (RMS)* is the core component for interaction with the other components and it also stores all data. The main functions of this component include:

- Manages and stores available world realizations and rooms per world (size, position, type of walls, etc.)
- Manages and stores available (supported) 3D objects and learning tools for each of the worlds, or rooms respectively
- Manages the creation, reuse and adaptation of learning settings for a specific world or room respectively
- Manages user and user groups as well as access rights for learning settings (configuration but also instances in-world)
- Manages room reservations (assign a pre-defined learning setting for a room and a learning group for a specific period of time)
- Provides interfaces for the RoomManager client and the virtual world realizations
- Manages the learning settings in the rooms of the virtual world realizations according to the room reservation information
- Stores learning tools to be used (states) in continued learning sessions (e.g. content of whiteboard)

The *RoomManager Client (RMC)* provides a graphical interface to the end-users to configure the learning settings of rooms and also to handle related administration such as user management and room reservation. For this prototype, the components have been designed as Web application with the following main functions:

- Manage all entities of the RMS components (user and user groups, learning setting creation and template re-use, room reservation)
- Makes the entities of the RMS components accessible
- Provides a graphical user interface for drag-and-drop learning setting creation, re-use and re-configuration

The development uses the Java Script Object Notation (JSON) client-server communication, which is based on the REST architecture. JSON enables a synchronized data

transfer between client and server and uses the persistence concept where all changes are stored on the server.

Both the RMS and RMC have been implemented in Java Standard Edition (Java SE) by using the Eclipse SDK as integrated development environment. In addition, the following frameworks and libraries have been used: Hibernate (storage and retrieval of Java objects by Object/Relational mapping); HSQLDB (Hyper Structured Query Language (SQL) Database in Java); Restlet (REST-ful Web framework for Java); Google Gson (Java library converting Java objects into JSON representations); Apache Log4J (logging and debugging); Apache Commons Configuration (generic configuration interface for Java); GWT (Google Web Toolkit (GWT) for building complex browser-based applications); gwt-dnd (drag and drop features for GWT).

B. OpenSimulator World Realization

Although the design of the RoomManager client and server supports to control of multiple virtual world environments, this research focuses on the realization based on the Open Simulator platform.

As an alternative to Second Life, *OpenSimulator (OS)* is an open source platform which is accessible by several clients and protocols. It partially supports features of Second Life and the corresponding 3D scenes, and it can also be connected by the Second Life client. OS is written in C# and can run on the Windows platform and on Linux distributions. OS instances can be connected to a grid of virtual world environments which can be administrated by various organizations. Each object in any of the OS instances has Universal Unique Identifier (UUID). Each of the servers can manage one or several logical sections which are called *regions*. Regions are managed in separated memory spaces. It can run the physics engine and scripts as well as manages objects in a scene graph and keeps observers updated (OSWiki, 2011a).

The OS server is designed to manage the 3D environment and has the following five main services: *User Server* manages the user authentication; *GridServer* manages UUIDs of regions and authenticates them in the grid; *AssetServer* manages assets and assigns UUID to each asset; *InventoryServer* links assets to users by means of UUIDs; *MessagingServer* handles the communication in-world such as who is allowed to listen to whose messages (OSWiki (2011b).

In order to extent the features and functionality of OS, two standardized mechanisms are offered to dynamically load components: *OS Application Plugins* implements the IApplicationPlugin interface and can be used to add functionality to the whole server. *OS Region Modules* implements the IRegionModule interface and provides additional functionality but also tools and 3D objects in world. Region modules are implemented in C# and can flexibly execute tasks such as creating and modifying objects, register and handle events, and communicate with external services (OSWiki, 2011c; OSWiki, 2011d).

The main functions of the *World Realization Component (WRC)* include:

- Supports configuration (instantiation and placement of 3D objects and learning tools) of several managed regions or learning spaces
- Connects and re-connects (after connection has been closed) to the RoomManager sever (RMS); each region can be connect to individual RMSs
- Receives commands from RMS and sends back notifications on state changes of managed tools
- Manages 3D objects and tools for each of the regions

The objective is to control learning settings respectively 3D object and learning tools in world. Previous experiences in Second Life on this issue have shown that the solution based on a Linder Script object has too many limitations (see also Gütl et al., 2010), thus we have decided to make use of the OS region module architecture. Development language for OS extensions is C#. As the development platform was Linux, the Mono development framework (Mono, 2013) is used to support C# on the Linux platform and MonoDevelop (MonoDev, 2013) as integrated development environment.

The *RoomManager Module (RMM)* represents the component in-world and is implemented as a *non-shared region module* for each available region (OSWiki, 2011d). The RMM is designed and implemented to support flexible layout and location of rooms which are created on demand. Each of the regions is assigned a scene object that enables access to other modules and the core functionality of the OS. Information to create learning settings in-world and states of the learning tools in-world are exchanged between the RMS and RMM over the communication channel (JSON messages are sent via a permanent socket connection). Thus, each region module connects to a server port and the data gets serialized and de-serialized by built-in C# and Mono mechanism.

To give a better idea how the RMM is working, the creation of a learning setting is illustrated: The RMM communication component *RoomManagerConnector* retrieves the relevant data of the learning space to be built. The *RoomBuilder* component creates and manages all objects in-world. In a first step the learning room is created according to the configuration of the RMS, i.e. walls, roof, floor, sliding door and room sign are created and texture properties are assigned to all objects accordingly. In the second step the *3D content objects* (such as tables and chairs) and *learning tools* are created out of an object repository and placed in-world accordingly. All objects are added to the scene object in the corresponding region of the OS. For each object a C# class is provided for creating the 3D instances in world.

Learning tools in-world can be seen as complex 3D object compounds which also provide ways for interaction and collaboration, such as presentation walls, whiteboards, and concept maps. Compared to simple 3D objects, the challenging issue of such tools is that during interaction the state changes; i.e. students add lines and words on the

whiteboard. In the event where a learning session is requested to be continued, such state information are required to be stored persistently and recalled for the next session, the learning setting is created. Two approaches have been tested: firstly, Linden Scripts (LS) have been added to the learning tools but no sufficient way has been found how to communicate state changes between LS and the region module. Secondly, the avatars interaction with the learning tools is directly communicated to the region module and the RMM updates the state of the learning tool by sending the state to the RMS. This second approach provides a solution to automatically send updates to the RMS for status updates and persistent storage.

C. Component Communication

Figure 2 provides an overview of the communication and interaction between the components following the conceptual architecture outlined in Figure 1.

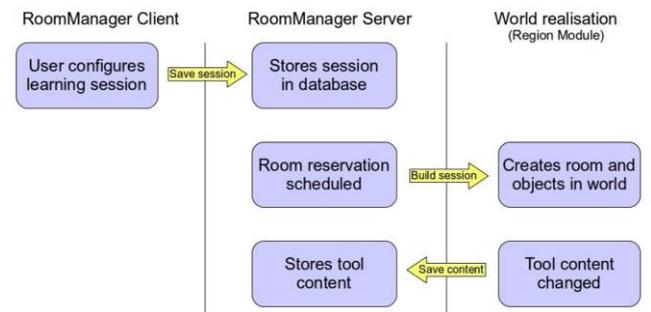


Figure 2. Overview of Interaction of Prototype Components (Haas, 2012)

Starting with the room manager client, it renders the graphical user interface to manage and store settings and information. By focusing on the management of learning settings, any information of changes is transmitted and stored in the component of the room manager server (RMS). The RMS manages all learning settings and schedules the instantiation of learning settings (room layout, object and learning tools) of the world realization and the OpenSim environment. Any changes to the learning tools or the status of the learning tool (including content) is sent back to the sever component of the room manager. This enables a recall of the state of the learning setting that allows reuse to scaffold learning from the previous state of the learning.

IV. USAGE ASPECTS OF THE PROTOTYPE

The aim of this section is to give a brief overview of the users' viewpoint. In Figure 3, the menu bar on the left gives an overview of the features. This includes user and user group management, virtual world and room management, the management of learning settings and room reservation. Figure 3 shows as one example the user management feature in detail. A major feature of the prototype is the room configuration support. Figure 4 shows the window of the setting editor. This editor enables users to define the room layout, select objects and learning tools and place them according to the specific needs, and manage access rights of users. Settings can be saved and offered to other users or

learning groups to book and use the space. Pre-existing settings can be adapted to create user's specific learning settings. Figure 5 depicts the instantiation of the learning setting in a OpenSim world. According to the access rights, the learning setting is public or restricted to a user group or to a list of users. Content changes of learning tools are persistently stored by the room manager and can be recalled for the next learning session. Figure 6 shows the room reservations feature of the prototype.



Figure 3. Room Manager Client provides a menu on the left for managing users, worlds and rooms and well as room reservations.

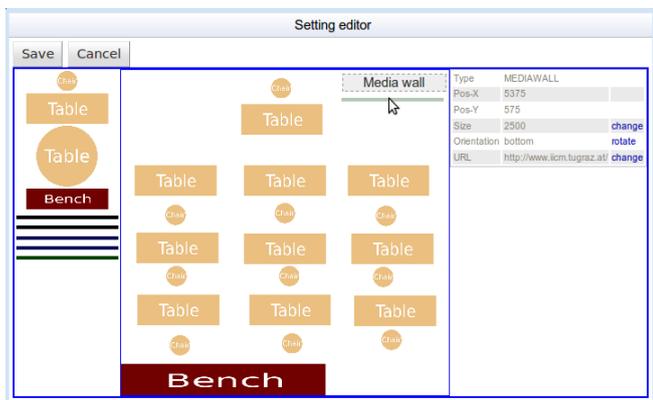


Figure 4. Room Manager enables end-users to reuse objects and learning tools and to configure learning settings by using 'drag and drop'.

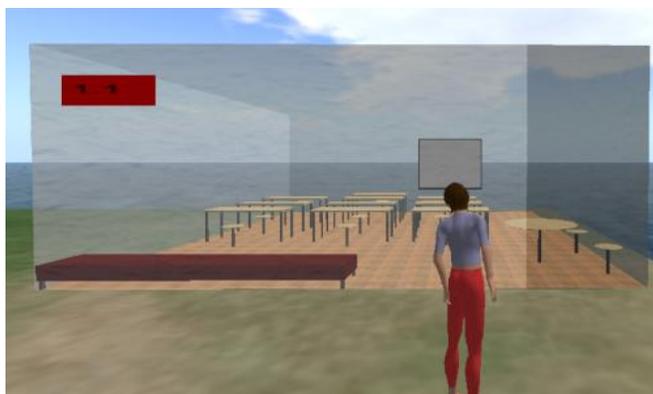


Figure 5. Example of learning setting instantiated in-world after specifying the setting in the room manager.



Figure 6. Room reservation for room booking with specific learning setting.

V. LESSONS LEARNED

Research and development of the prototype for managing flexible learning settings have revealed a number of findings. Details can be found in Haas (2012), and the remainder of this section gives a selection of findings.

The OpenSim (OS) platform has proven to be applicable and the project is in an early stage of development. The plug-in mechanism of the OS platform for controlling and interacting with objects in-world has proven to be a sufficient approach.

The room manager has followed a design which is flexible to support various virtual world platforms and a number of rooms and regions for the virtual worlds. OpenSim however has the limitation, where regions are limited to a maximum of 256 meters in length and width. For larger regions and overlapping regions a shared region module is required. The extension of further learning and collaboration tools is simple to build and this requires tools data to be added with an icon at the room manager. To ensure recall and reuse, another class must be added to the region module managing the creation and state of the tool in-world. The current version of the region module is not tracking position changes of the managed tools (learning and communication tools) and 3D objects. A possible solution for the managed tools is to have regular monitoring of the position and occurrences of changes are notified to the room manager.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have reported on various ways in both virtual and traditional approaches to support new forms of learning requiring flexibility and adaptation of learning groups' needs. We have also emphasized the need for easy configuration, adaptation and reuse of learning settings in virtual immersive environments. A promising proof of concept for the Second Life platform has led to an improved approach for the open source platform using OpenSimulator.

The basic idea is to design and develop a simple tool for end-users to create, manage, reuse and adapt learning settings by pre-existing 3D objects and learning tools by a simple 2D drag and drop interface. In order to efficiently use IT and other resources, a room manager enables users to book learning rooms with specific learning settings and the settings are instantiated -world for a specific time frame. The main contribution of this paper is to introduce the conceptual requirements for a flexible room management tool and a first prototype realized as Web-based server-client application capable of controlling different 3D world environments.

For future work, our focus on the research is on standardized interfaces and a communication language as well as a linked ontology to manage regions, rooms and learning settings as well as 3D objects and learning tools. We also want to support several virtual world platforms such as Open Wonderland and Unity. Further work needs to focus on approaches to keep learning settings up-to-date and incorporating changes that are made (eg following group discussion or decision) in-world. Also, we aim to incorporate gamification aspects and possibly integrate a flexible assessment component. As a specific application scenario, we would like to apply this research for skills training settings in the logistic domain as part of the nDIVE (multi or n-Dimensional Immersive Virtual Environment) project.

ACKNOWLEDGMENT

This paper is part of research resulting of visiting academic activities of Karl Haas and Christian Gütl at the School of Information Systems, Curtin University. The visits were supported and sponsored by School of Information Systems, Curtin University, Perth, Australia and the Institute for Information Systems and Computer Media (IICM) at Graz University of Technology, Graz, Austria.

This paper is also linked to aspects of the research project nDIVE (<http://ndive-project.com>). Support for the production of this publication has been partly provided by the Australian Government Office for Learning and Teaching (Development of an authentic training environment to support skill acquisition in logistics & supply chain management, ID:ID12-2498). The views expressed in this publication do not necessarily reflect the views of the Australian Government Office for Learning and Teaching.

REFERENCES

[1] Chang, V., Gütl, C., Kopeinik, S., and Williams, R. (2009). Evaluation of Collaborative Learning Settings in 3D Virtual Worlds. *International Journal of Emerging Technologies in Learning (iJET)*, Vol 4 (2009): Special Issue: ICL2009, 6-17.

[2] Chang, V., and Gütl, C. (2010). Generation Y Learning in the 21st Century: Integration of Virtual Worlds and Cloud Computing Services. In Z. Abas et al. (Eds.), *Proceedings of Global Learn Asia Pacific 2010* (pp. 1888-1897). AACE, Penang, Malaysia, May 17 - 20, 2010.

[3] Daniel, J. (2012). Making Sense of MOOCs: Musings in a Maze of Myth, Paradox and Possibility. *Journal of Interactive Media in Education*, 2012, 18, <http://jime.open.ac.uk/2012/18>.

[4] de Freitas, S. (2008). *Serious Virtual Worlds. A Scoping Study*. Prepared for the JISC e-learning programme, last edited November

2008, <http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf>. Accessed 17 Feb. 2010.

[5] Gütl, C., and Chang, V. (2008). "Ecosystem-based Theoretical Models for Learning in Environments of the 21st Century". *International Journal of Emerging Technologies in Learning (iJET)*, submitted Nov. 2008, December 2008.

[6] Gütl, C. (2008). *Enhanced Computer-based Support for Personalized Learning Activities*. Post Doctoral Lecture Thesis. Graz University of Technology, Austria.

[7] C. Gütl. (2011). "The Support of Virtual 3D Worlds for enhancing Collaboration in Learning Settings", in Francesca Pozzi and Donatella Persico (Eds.) ,*Techniques for Fostering Collaboration in Online Learning Communities: Theoretical and Practical Perspectives*, IGI Global, 2011, 278-299.

[8] Gütl, C., Chang, V., and Freudenthaler, S. (2010). How to Support More Flexible Learning Settings in Second Life. In *Proc. of the ICL 2010*, Hasselt, Belgium, September 2010 129-141.

[9] Haas, K. (2012). *Flexible Learning Environment in Virtual 3D Worlds*. Unpublished master thesis, Graz University of Technology, 2012.

[10] Henckel, A., and Lopes, C. V. (2010). Stellarsim: A plug-in architecture for scientific visualizations in virtual worlds. In O. Akan et al. (Eds.), *Facets of virtual environments* (Vol. 33, p. 106-120). Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-11743-5_9.

[11] Kaplan, J. and Yankelovich, N. (2011). "Open Wonderland: An Extensible Virtual World Architecture," *Internet Computing, IEEE* , vol.15, no.5, pp.38,45, Sept.-Oct. 2011, doi: 10.1109/MIC.2011.76.

[12] Leidl, M., and Röbling, G. (2007). How will future learning work in the thirddimension? *SIGCSE Bull.*, 39 , pp. 329-329. <http://doi.acm.org/10.1145/1269900.1268897>.

[13] Mono (2013). Main page. Mono Website, http://www.mono-project.com/Main_Page. Accessed 15 April 2013.

[14] MonoDev (2013). Main page.MonoDevelop from <http://monodevelop.com/>. Accessed 15 April 2013.

[15] Moro, A., Mumolo, E., and Nolic, M. (2010). Building virtual worlds by 3d object mapping. In *Proceedings of the 2010 acm workshop on surreal media and virtual cloning* (pp. 31-36). New York, NY, USA: ACM. <http://doi.acm.org/10.1145/1878083.1878092>

[16] OSWiki (2011a). Main page. OpenSimulator Wiki, http://opensimulator.org/wiki/Main_Page. Accessed 02 April 2013.

[17] OSWiki (2011b). Configuration. OpenSimulator Wiki, <http://opensimulator.org/wiki/Configuration>. Accessed 02 April 2013

[18] OSWiki (2011c). How to create a dynamic plugin. OpenSimulator Wiki, 02.04.2013, from [http://opensimulator.org/wiki/How to create a dynamic plugin](http://opensimulator.org/wiki/How_to_create_a_dynamic_plugin). Accessed 02 April 2013

[19] OSWiki (2011d). Region Modules. OpenSimulator Wiki, <http://opensimulator.org/wiki/RegionModules> Accessed 02 April 2013

[20] Quintella, F., Soares, L. P., and Raposo, A. B. (2010). DWeb3D: A toolkit for developing x3D applications in a simplified environment. In *Web3D'10*, pp. 45-54.

[21] Reiners, T., Wood, L., Chang, V., Gütl, C., Herrington, J., Teräs, H. and Gregory, S (2012). Operationalising Gamification in an Educational Authentic Environment. In: *Kommers, P., Issa, T., Isaías, P. (Eds.), IADIS International Conference on Internet Technologies & Society 2012*. Presented at the IADIS 2012, Perth, Australia, pp. 93–100.