

How to Support More Flexible Learning Settings in Second Life

Christian Gütl^{1,2}, Vanessa Chang², Stefan Freudenthaler^{1,2}

¹ Institute for Information Systems and New Media (IICM), TU-Graz, Austria

² School of Information Systems, Curtin University, Perth, WA, Australia

Keywords: *Virtual 3D World, Second Life, E-Learning 2.0, Flexible Learning Environment, Customization*

Abstract:

Together with advanced technologies, learning environments in today's education context are becoming more and more sophisticated. The learning requirements and options for students and instructors are also changing with different types and combination of learning modes (face-to-face, online learning, and blended learning). In addition, students are also located at different geographical regions and these students have to be given the opportunity to choose a mode of learning that suit their lifestyle and location. Virtual 3D Worlds can support such options by taking advantage of the multiple communication channels and the social engagement associated with presence and awareness. There exist a number of learning settings which are designed for traditional learning approaches that do not support flexible environment. This situation has motivated us to initiate research towards a flexible system which supports configuration and adaptation of virtual learning environment according to the learners' needs. In this paper, we will present the concept of reuse of resources which were invested to support flexible learning settings. Preliminary findings based on the virtual world Second Life are also discussed.

1 Introduction

In today's knowledge driven and globalized world of the 21st century, it is becoming a bigger challenge to plan and implement adequate formal and vocational education approaches. The main stakeholders in this process are confronted with a variety of issues given by the ever-changing political, economical, social, technological, environmental and legal situations. More importantly, learning groups of students and instructors are becoming increasingly dynamic in terms of their goal aspirations and actual needs. Learning groups also tend to be geographically dispersed [11]. Past research had reported that geographically distributed learners had negative learning experiences (eg. isolation, loneliness, lack of community) with distance learning. It was acknowledge that only sufficient support of information and communication technologies (ICT) can enable to meet these requirements for distance education.

Continuing with a historic view on the notion of learning, educational researchers have treated the learning process as purely psychological phenomenon in the past. During the 20th century, philosophy as well as education and cognitive science researchers have introduced new views on learning and knowing, where contextualization, social aspects and interpersonal interactions had become important. As one of the emerging theories, collaborative learning has gained increasing interest over the past two decades [1; 30]. By narrowing down to collaboration in the learning process, the main interest for many years in research had focused on the individual learning. Until the beginning of the early 1980s,

cognition has mainly been seen as the result of the individual information process and the context of social interaction as the environment for individual activities. As the interest in collaborative learning phenomena grew and different approaches such as the socio-constructivist, socio-cultural, and shared cognition emerged. Collaborative learning activities apply different approaches in-class or out-of-class which range from classroom discussions to group-based assignments. Collaborative learning can involve students more actively as well as stimulate social skills and interpersonal skills [10; 28; 29].

Pedagogical and cognitive science developments as outlined above have consequently influenced developments of learning technology. Thus, technologies for distance education have evolved from tools mainly used for organizing and delivering learning content in the past towards enhanced applications which strongly support communication and collaboration. As indicated previously, different studies such as Bernard, Rojo de Rubalcava and St-Pierre [2], suggest that distance learning have problems of high dropout rates and low level of quality learning. Other reasons include the feeling of isolation, procrastination, lack of multiple communication channels, and difficulties related to self-regulation. Although there are collaborative tools to overcome these problems, a great number of pre-existing approaches and technologies have deficiencies related to the interpersonal communication perspective. Virtual 3D worlds, such as Second Life or Project Wonderland, offer an opportunity to either mitigate or overcome these issues [12].

From a collaborative learning viewpoint, the advantages of virtual 3D worlds include multiple communication channels, including the feeling of presence and awareness. Such learning environments can reduce barriers between group members, support the feeling of belonging to a community, and facilitate collaboration between group members [12]. According to De Freitas [8], there exist a wide range of different virtual world applications and packages which were designed for training and learning in various subject domains including activities such as virtual lectures and seminars, pre-work training scenarios, role plays, and group assignments. Virtually all of them are designed to support specific pedagogical approaches and learning objectives, and therefore do not support the concept of reuse of invested resources. Hornik, Johnson and Wu [14] emphasized in this context there is a *“need to flexibly design VLEs (Virtual Learning Environments) such that the technology support of learning is flexible and adaptable to match users’ learning conception”*.

The situation outlined so far has motivated us to initiate research towards a flexible system which supports configuration and adaptation of virtual learning environment according to the learning communities’ specific needs. In this paper we focus our preliminary findings based on the virtual world Second Life. The structure of the remainder of this paper is organized as follows: Section 2 gives an overview of an analysis of the variety of educational aspects and their implications for virtual world based learning environments, followed by related work on flexible approaches in Section 3. Section 4 discusses requirements for a flexible solution and outlines design and development aspects, followed by a usage viewpoint and finding in Section 5.

2 Situation Analysis

Learning and teaching processes are highly complex and they are influenced by a number of aspects and characteristics, involving various stakeholders and instantiated by a variety of approaches. To illustrate this, the IMS Learning Design Specification defines a model for tasks and activities, assignment to roles, and the workflow of a unit of learning. Each unit of learning is characterized by learning objectives (achievements of the unit), roles (different involvement to achieve the objective either as learner or other stakeholders),

activities accordingly to the role either learning activities or support activities, and environment including the learning content and the physical or virtual ‘space’ learning [5]. There are also three widely accepted learning models which focus on different aspects: (1) objective model (transferring objective knowledge from a knowledge expert to a domain subject novice), (2) constructivist model (individuals discover knowledge and incorporate the knowledge in the knowledge structure), and (3) collaborative model (extends the constructivist model by creating a shared understanding). The learning model including instructor characteristics, learner characteristics and technology, play key roles in achieving successful learning outcomes [14]. Also learning communities are becoming increasingly dynamic in terms of learning group formation, and their specific learning requirements [11]. This situation uncovers the challenge that technology needs to be flexible to support a variety of learning situations and settings. Often technological solutions are not flexible enough, thus leading to compulsory learning processes where the learners must follow and this might have negative effects on the learning outcomes [14].

By further focusing on technology-based distance learning, Scopes [23] defines the concept of Cybergogy as *“learning experiences that are cognitively, socially, and emotionally engaging for distance students with diverse background with learner engagement at the core”*. In this context virtual 3D worlds (V3DW) have become increasingly popular and more widely accepted within the last few years. V3DW characteristics include learning control, collaboration, persistence, 3D interaction and experiences, immersion and interactivity [8; 12]. Popular examples of such virtual worlds are Second Life [16], OpenSim [18] and OpenWonderland [20]. Scopes [23] clearly points out the importance of V3DW *“to supplement blended learning and to bridge the gap between asynchronous online course participation and physical classroom experience”*. However, V3DW must accomplish the required but also perceived learning objectives of teachers, students and the other stakeholder in the learning processes (Jones & Warren, 2008). Coffman and Klinger (2008) argue that these requirements may facilitate by ‘rich environments’ which *“offer many different types of learning activities to appeal to different learning approaches.”* For providing such great flexibility in learning settings, an easy composition of tools is required which supports (a) knowledge acquisition by direct instruction or interactive exploration, (b) scaffolding the students’ ability, (c) assessment and feedback, (d) conversation and collaboration, and (e) social or contextual support. [11; 23]

The situation analysis reveals the importance of virtual learning environments which are highly customizable, provide sets of tools but also facilitate the integration and easy adaptation of pre-existing tools. Moreover, the learning communities should be able to easily define the specific virtual learning environment and/or re-use predefined settings. Although there exists a variety of V3DW-based learning tools and environments [8], there is a lack of attention on the required flexibility.

3 Related Work

From the viewpoint of 3D learning content creation, there are external tools which support stakeholders to design the learning processes. Free software such as Google SketchUp and Blender [3; 10] or commercial products such as Cinema 4D and 3Ds Max are available. For the purpose of ‘reuse’ of (learning) objects, it is recommended to use 3D repositories such as Google 3D Warehouse [9] and Blender Model Repository [3].

Most prominent virtual worlds, such as Second Life and OpenSim have built-in tools to create objects ‘in world’ and they also provide scripting languages and application

programming interfaces (API) to make objects interactive, and additional applications can also be built [22]. Over the last few years, users have started to share and reuse content and tools, and this practice is becoming common with the establishment of a real business on virtual ‘goods’ such as the Second Life Market Place [24]. Learning objects and tools are also being shared and re-used [27]. Similar services have also emerged for OpenSim and Second Life goods can also be imported. From the viewpoint of flexible creation of learning settings, the concept outlined thus far requires expertise and effort to find and compile content and tools constituting the appropriate settings. The complexity increases where learning settings requires different roles and access rights. Moreover, scripting language and API support are limited and this may lead to serious implications on features and functionality of developed applications.

To overcome the problems – at least partly, advanced approaches and tools have been developed. The OpenSim project offers a Region Module approach [18] which enables developers to enhance the system by developing specific functionality. New developed and pre-existing modules can be added to the system by deploying the module and editing a configuration file [19]. OpenWonderland supports a Web-based tool for adding and managing modules. These modules can be activated and instantiated ‘in world’ by the users [21]. The built-in feature to share desktop applications and to drag-and-drop content ‘from and into the world’ can also support the set-up of flexible learning environments. Maher, Rosenman and Merrick [17] have applied their approach in a distributed agent system to import external model to create different ‘in world’ representations for multi-disciplinary viewpoints and to keep track of the ‘in world’ or ‘out world’ updates of the representations or the model.

There are also specific tools and frameworks which aim to ease the creation of flexible learning environments. Based on DXStudio, a 3D game creation and edition platform, the ALIVE suite supports educators to create their own 3D environments for learning and training purposes [7]. In Second Life, instructors can use the Toolkits Project that provides ‘in world’ inventory to create their own learning environment. The inventory includes a chat box, an action board, a feedback survey tool, a chat logger, and a notecard dropbox [13]. Simulation Linked Object Oriented Dynamic Learning Environment (SLOODLE) is a toolset to link Second Life and Moodle learning management system. It supports authorization of users between the systems and enables information exchange between the two systems by controlling the access rights. ‘In world’ support for creating flexible learning applications include chat tool, quiz tool, assignment dropbox, tool for object access and collaborative website browsing as well as an award system to enhance student engagement [15]. A similar project also exists to integrate Moodle into OpenSim [6]. Although Service Oriented Architecture (SOA) is a powerful approach for building flexible learning environments, this has not yet been fully exploited in virtual world research and development. Booth and Clark [4] report on a prototype of a pluggable services-oriented learning environment which makes use of Project Wonderland’s shared applications feature to make learning tools accessible ‘in world’. Cai, Sun, Farh and Ye [5] introduce a framework to build flexible learning environment in different virtual 3D worlds which underpinned by the idea of an open, flexible service ecosystem that includes virtual learning services. Finally, there is also research on dynamic creation and instantiation of virtual learning spaces driven by modern architectural design methods, such as Willkomm [32].

4 A First Proof of Concept

The situation analysis and the related work of flexible learning environment built on virtual 3D World technology have motivated us to initiate a collaborative research project between School of Information Systems (SIS), Curtin University, Australia and Institute of Information Systems and Computer Media (IICM), Graz University of Technology, Austria. In the first stage of the research project it was decided to design and develop a proof of concept based on Second Life [16]. The requirements on an abstract level include (a) easy to configure, store, and reuse learning settings facilitating the actual needs of specific learning communities, (b) intuitive user interface to support professional learning designer, teachers and students to define their required settings, (c) provide a mechanism to extend the library of learning tools and inventory of 3D objects, and (d) simple instantiation of learning spaces 'in world' by learning communities incorporating the management of user and group rights as well as the storage and recall of such spaces including states and content of tools for further learning sessions.

4.1 Development Viewpoint

The conceptual architecture as shown in Figure 1 and the technologies are outlined for the proof of concept. The requirements include the following design decisions for the development of the proof of concept. In general, the project is separated into two parts:

1. Web application:
 - a. A graphical user interface, accessible with a web browser, as an intuitive platform to manage dynamic learning environments.
 - b. A database to store data and information about the available Second Life items and the developed learning scenarios.
 - c. A backend interface to communicate between the database and Second Life
2. Second Life :
 - a. A central object to handle the data exchange between the Web application and Second Life platform.
 - b. The single Second Life object executes the received data in order to instantiate or update learning environments; this includes the instantiation and placement of other Second Life objects (inventory and learning tools).

Web Application

The Web application is divided into three components: the graphical user interface (GUI), the database layer, and the backend interface.

The *Graphical User Interface (GUI)* enables users to define and update learning settings. The GUI component is used as dynamic Web page to enable users to interact with the application in a convenient and simple way. To develop the application, Google Web Toolkit (GWT) version 2.0.3 was chosen as the framework with Eclipse as an editor. GWT is an open source set of tools that allows web developers to create and maintain complex JavaScript front-end applications in Java [31]. The main reason to use GWT was the possibility to use asynchronous remote procedure calls (RPC) and the support available for dynamic and reusable UI components for drag and drop.

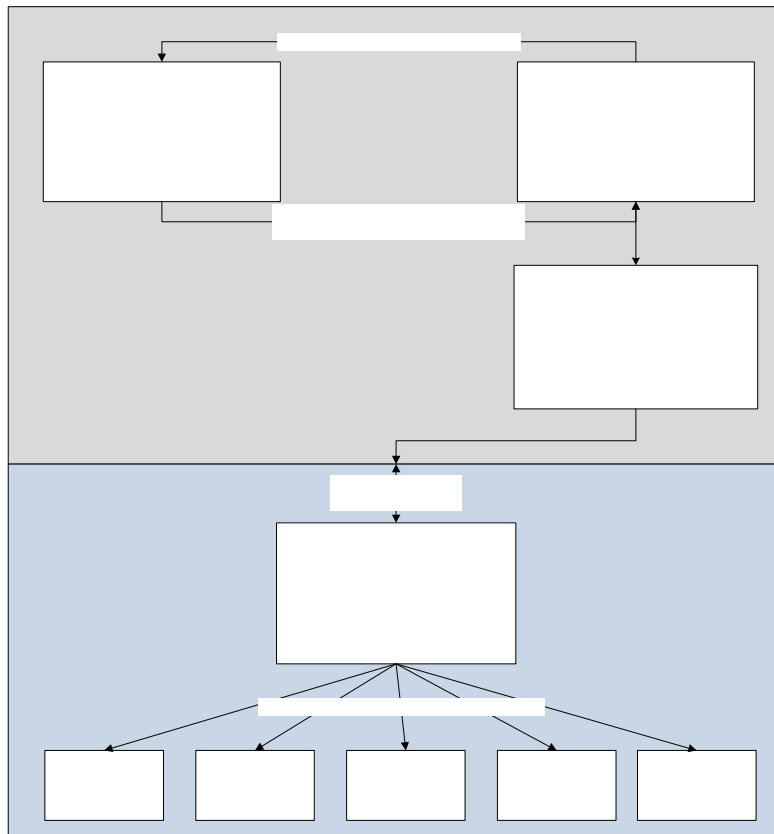


Figure 1 – Conceptual Architecture

User I
manage lea

The *Database Layer* provides the data store for the application which was built on relational database on MySQL version 5.1.

The *Backend Interface* handles the data exchange between the Web application and the Second Life platform. The interface is used in the proof of concept as a PHP script and can be polled from Second Life by means of HTTP requests. A data structure has been defined to create and control the objects in Second Life (see

Figure 2).

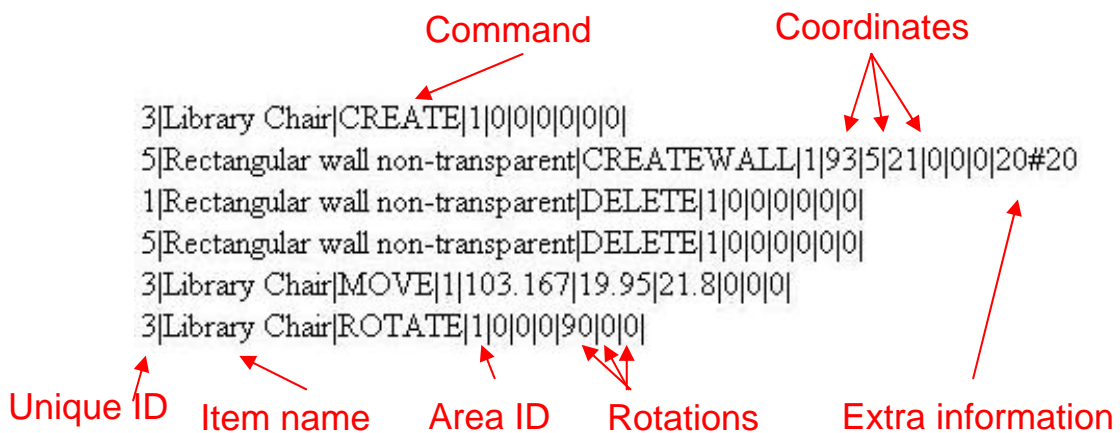


Figure 2 - Response of the background interface after a request from the 'SL Brain'

SL Brain and 'In-World' Object Control

In order to create and control the learning settings within Second Life, the design decision is to implement the functionality by one central object which is known as the Second Life Brain (SL Brain). This object is realized as a simple cube 'in world' which includes the use of Linden Script Language (LSL).

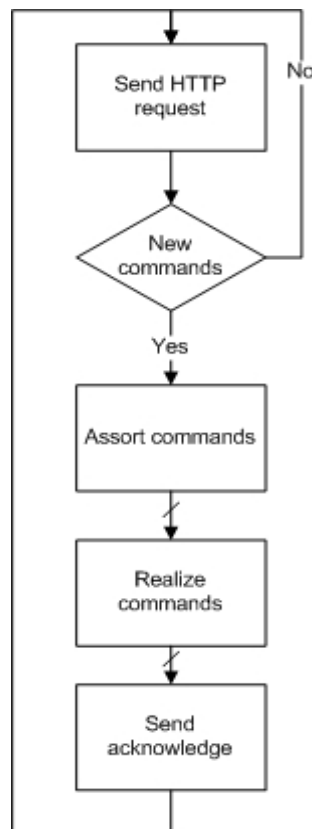


Figure 3 – ‘SL Brain’ procedure

The script as shown in

Figure 3 runs in an infinite loop. A “HTTP request” is sent regularly to the backend interface of the Web application. It is necessary to wait a certain time between each passes because Second Life limits the amount of connections to external servers. After receiving several commands, the script assorts the different orders. Currently the following commands are executed by the SL Brain:

- CREATE
- CREATE WALL

If the ‘SL Brain’ receives a CREATE command, the dependent Second Life object (every item of a learning setting is created this way) will be ‘rezzed’ next to the ‘SL Brain’.

Rezzing is the LSL command for creating an object. It must be kept from the root object in the inventory to use this function. An important part at this phase is that the ‘SL Brain’ will hand-over a unique ID, which it receives from the GUI and is unique in Second Life and in the web application.

Figure 2 shows a parameter to create an object. This ID will be the channel where the new Second Life object can ‘listen’ for more commands from the ‘SL Brain’. In Second Life, LSL scripts are able to communicate with each other with the ‘listen’ command [25]. This method requires a channel to ‘listen’ to other messages from other LSL scripts. Thus, the unique ID supports every item of the environment setting by receiving its own commands from the ‘SL Brain’. The CREATE WALL command creates the basic structure of the

learning setting, such as walls, floor and roof. These components consist of several single components, because Second Life limits the size of a single object to 10 meter in width, height and length. Therefore, the ‘SL Brain’ has to move to the position before it ‘rez’ the single objects. The reason for this is that objects can only be rezzed up to a radius of 10 meter away from the root object.

All other Second Life objects, except the ‘SL Brain’ of this project, have to have a special LSL script included. Otherwise it cannot be controlled by the ‘SL Brain’. This script is responsible to execute the following commands, which has been forwarded from the “SL Brain”:

- MOVE
- ROTATE
- DELETE

These commands, as illustrated in

Figure 4, shows the flow of data between the background interface of the web application of the ‘SL Brain’ and a Second Life object. An object is created by pressing the ‘Send to SL’ button.

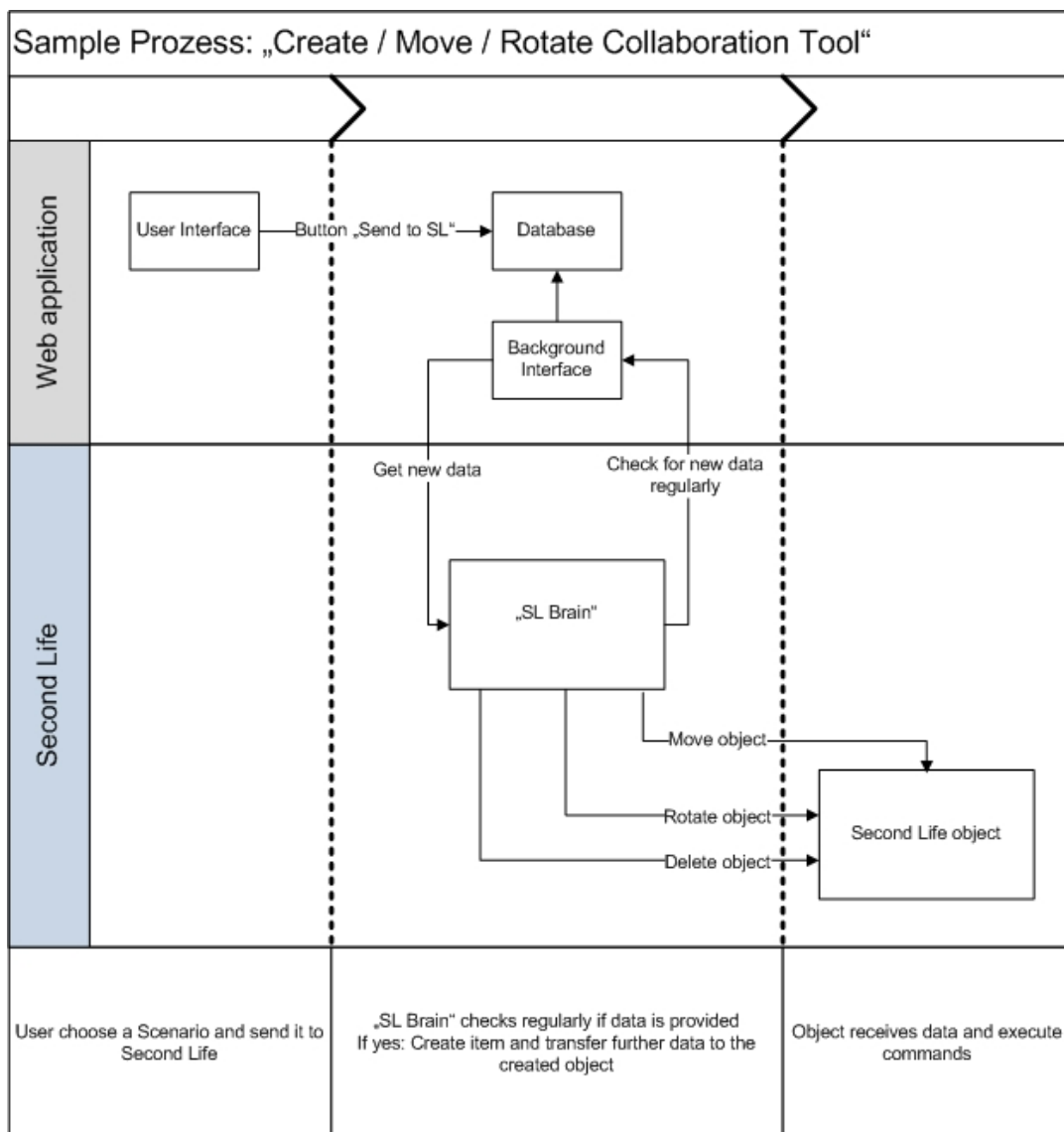


Figure 4 Sample sequence

4.2 A Glimpse from the Usage Viewpoint

The GUI is built with a simple and intuitive concept as shown in Figure 5. A user chooses a Second Life area where the learning environment should be built. A corresponding scaled eagle’s 2D eye view is shown in the middle of the area plus other menu controls. The menu on the right offers Second Life objects, such as collaboration and presentation tools, inventory and the basic look of the surroundings, and the ability to move via drag and drop into the scenario area. In the bottom menu, objects can be rotated or deleted and scenarios can also be saved or loaded. The ‘Send to SL’ button transfers the data from the selected scenario into the database. Coordinates, positions, and sizes are converted into usable Second Life dimensions. At the next polling request of the ‘SL Brain’, the script handles the data and updates the setting accordingly ‘in world’. As a future work we planned to combine saved scenarios to timeslots as well as user management including user and group rights. With this, it would be possible that the ‘SL Brain’ disposes scenarios for certain time spans. Figure 6 shows the instantiated learning environment in Second Life.



Figure 5 – Screenshot of the graphical user interface



Figure 6 Realized learning setting in Second Life

5 Findings and Recommendations

5.1 Implementation viewpoint

From the implementation view there are several positive aspects and obstacles. On the positive side, it was simple and convenient to develop web applications with Google Web Toolkit. The framework is easy to learn and the community is well represented. There are a number of websites supports (such as Second Life Wiki [25]) where programmer can find quick answers. The important advantages of GWT was the possibility of asynchronous remote procedure calls, the easy handling of drag and drop and that GWT uses JAVA. The development in Second Life with LSL is simple to learn for a skilled programmer. It is easy to get used to LSL and the programming in the virtual reality. There are also many ‘in world’ places where one can learn to program in Second Life.

The development in Second Life has also shown some problems and restrictions:

- Usually a programmer develops a web application and database on a local computer and transfers everything to a public server upon finishing. In this process it is possible that the Web application is not accessible from the public Internet, because the developer does not have a public and static IP address or is behind a firewall. As the communication initiated by Second Life objects require Internet access within the official IP range of the Web application, some workaround is required.
- Programming in Second Life, especially object movements, can be challenging as well. If the program moves an object to an incorrect Second Life coordinates, the prim will disappear and almost untraceable. Usually a restriction of 10 meter limits this, but in our script objects makes stopovers every 10 meters to avoid this error.
- Unfortunately it is not possible to get access to prims without including scripts into them. That means every single objects need a LSL script and a unique ID to be controlled by the ‘SL Brain’. Equally it has to be added in the online database with their exact dimensions and description. That produces a lot of work for the developer.
- Every item and avatar in Second Life has an own collection of SL objects, called inventory. If a prim want to be accessed or ‘rezzed’ by another prim, it has to be kept in the inventory of the root object.
- Another limitation in Second Life is the number of prims per parcel and that only one media is allowed per parcel. This has the effect that the GUI limits the amount of collaboration tools, which uses media, to one per area.
- Further LSL methods, especially the method to rez objects produces time delays [26].

- The trading system in Second Life offers several opportunities of copyright for the author. It is possible to buy objects with locked scripts. That prevented the necessary option to include the script, which is able to communicate with the 'SL Brain'.

5.2 Usage viewpoint

The project is still in a prototype and infancy phase. The initial findings from assessment of the proof of concept are outlined within this section.

The Web application is intuitive and easy to operate. There are also known problems that include the following:

- The GUI pop-up menus do not show everything immediately. That can be a problem if the user uses lower screen resolutions or more functions will be added. For future work, a simple general menu has to be included.
- No help messages to avoid errors.
- The user interface does not show a status from the transfer into Second Life.

On the side of Second Life the current design decision leads to the following problems: if the connection between SL and the web is broken or a LSL scripts delays for too long, the commands for the "SL Brain" can be lost. This kind of error are not getting absorbed or detected as there is only a one way connection between the 'SL Brain' and the objects. That means the 'SL Brain' transfers the commands to the controlled objects, but does not check if the command was executed.

6 Conclusions and Future Work

In this paper, we have described a first proof of concept of a technology to support learning in a virtual learning environment. The proof of concept in Second Life virtual learning environment is flexible and adaptable to match the instructors' learning design and the students' learning styles. The GUI system which is accessible via a web browser is intuitive and simple to use. A reusable library of learning tools and inventory of 3D objects are available for users to choose from. The system is easy to configure, store, and learning settings can be re-used to facilitate the actual needs of the learning communities.

The current prototype is focused on the transfer of information from an external media into Second Life. A future feature of this project is to design a bi-directional communication between the Web application and Second Life. This will open up many possibilities for new features which will include the following:

- Implement more commands to create and manage the presentation tools, for example set up URLs, upload slides or movies via the website for collaboration tools.
- Transfer scenario and avatar status to the web application.
- Get feedback from the scenarios.
- Prepare user rights and access to the scenarios via the GUI.
- Get information and positions from external objects (not objects from the 'SL Brain').
- Include a zoom option into the web application to positioned objects more accurate and to manage bigger areas.
- Automatic recording of new objects into the database. The 'SL Brain' can send unknown objects with their dimensions and properties to the Web application, however, it is still a manual process to include a new object via a LSL script and save to the 'SL Brain' inventory.

- Develop the GUI as a 3D model to be more intuitive.

The situation thus far has motivated us to initiate further research to realize an adaptable and flexible system to enhance learning which is accessible by the learning communities.

References

- [1] Alavi, M., & Dufner, D. (2005). Technology-mediated collaborative learning: A research perspective. In S. R. Hiltz & R. Goldman (Eds.), *Learning together online: Research on asynchronous learning networks* (pp. 191-213). Mahwah, New Jersey: Lawrence Erlbaum.
- [2] Bernard, R.M., Rojo de Rubalcava, B., & St-Pierre, D. (2000). Collaborative online distance learning: Issues for future practice and research. *Distance Education*, 21(2), 260-277.
- [3] Blender (2010). Blender Model Repository. Official Blender website, last retrieved Feb. 15th, 2010, from <http://e2-productions.com/repository/>
- [4] Booth, A. G., & Clark, B. P. (2009). A service-oriented virtual learning environment. *On the Horizon*, 17, 3, 232 – 244.
- [5] Cai, H., Sun, B., Farh, P., Ye, M. (2008). Virtual Learning Services over 3D Internet: Patterns and Case Studies. *IEEE International Conference on Computing Services (SCC 2008)*, pp. 213-219.
- [6] CTN (2009). E-Learning platform Moodle integrated to Opensim. *CyberTech News*, last edited April 3rd, 2009; last retrieved Feb. 15th, 2010, from <http://www.cybertechnews.org/?p=1383>
- [7] de Byl, P. (2009). Making Web3D Less Scary: Toward Easy-to-Use Web3D e-Learning Content Development Tools for Educators. *Innovate: Journal of Online Education*, 5, 5 Jun-Jul 2009, last retrieved Feb. 15th, 2010, from http://innovateonline.info/pdf/vol5_issue5/Making_Web3D_Less_Scary_-_Toward_Easy-to-Use_Web3D_e-Learning_Content_Development_Tools_for_Educators.pdf
- [8] De Freitas, S. (2008). Serious virtual worlds. A scoping study. Prepared for the JISC e-learning programme, last edited November 2008, last retrieved Feb. 17th, 2010, from <http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf>
- [9] Dillenbourg, P., Baker, M., Blaye, A. & O'Malley, C. (1996). The evolution of research on collaborative learning. In E. Spada & P. Reiman (Eds) *Learning in Humans and Machine: Towards an interdisciplinary learning science*. (Pp. 189-211). Oxford: Elsevier.
- [10] Google (2010). Google 3d Warehouse. Official Goolge website, last retrieved Feb. 15th, 2010, from <http://sketchup.google.com/3dwarehouse/>
- [11] Gütl, C., & Chang, V. (2008). Ecosystem-based Theoretical Models for Learning in Environments of the 21st Century. *International Journal of Emerging Technologies in Learning (IJET)*, Vol 3 (2008), 50-60.
- [12] Gütl, C. (2010). 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, 2010 (accepted)
- [13] HEA (2010). Second Life Toolkits Project. The Higher Education Academy, last retrieved Feb. 15th, 2010, from <http://www.ics.heacademy.ac.uk/resources/rlos/chandler/files/index.html>
- [14] Hornik, S., Johnson, R. D., & Wu, Y. (2007). When Technology Does Not Support Learning: Conflicts Between Epistemological Beliefs and Technology Support in Virtual Learning Environments. *Journal of Organizational and End User Computing*, 19, 2, 23-46.
- [15] Kemp, J. W., Livingstone, D., & Bloomfield, P. R. (2009). SLOODLE: Connecting VLE tools with emergent teaching practice in Second Life. *British Journal of Educational Technology*, Volume 40 Issue 3, Pages 551 – 555.
- [16] Linden Labs (2009). Second Life – What is Second Life? Linden Lab, San Francisco, USA, last retrieved August 22nd, 2009 from <http://secondlife.com/whatis/>
- [17] Maher, M. L., Rosenman, M., & Merrick, K. (2007). Agents for multidisciplinary design in virtual worlds. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 21, 3, 267—277.
- [18] OpenSim (2010). Getting Started with Region Modules. Open Simulator Project, last edited Feb. 5th, 2010; last retrieved Feb. 12th, 2010, from http://opensimulator.org/wiki/Getting_Started_with_Region_Modules
- [19] OpenSim (2010b). Open Simulator GForge – Projects and Modules. Open Simulator Project, last retrieved Feb. 12th, 2010, from <http://forge.opensimulator.org/gf/project/>
- [20] OpenWonderland (2010). Project Wonderland official website, last retrieved Feb. 15th, 2010, from <http://www.projectwonderland.com/>

- [21] OpenWonderland (2010b). Module Warehouse. Project Wonderland official website, last retrieved Feb. 15th, 2010, from http://www.projectwonderland.com/index.php?option=com_docman&cat=add_ons&Itemid=123
- [22] Scheucher B. (2010). Remote physics experiments in 3D virtual environment. Master thesis; Graz University of Technology, Graz, Austria, Jan. 2010
- [23] Scopes, L.J.M. (2009) Learning archetypes as tools of Cybergogy for a 3D educational landscape: a structure for eTeaching in Second Life. University of Southampton, School of Education, Master Thesis.
- [24] Second Life (2010). Second Life Market Place. Linden Labs, last retrieved Feb. 15th, 2010, from <http://secondlife.com/whatis/marketplace.php>
- [25] Second Life Wiki (2008). Listen. Linden Research, Inc. last retrieved Aug. 24th, 2010, from <http://wiki.secondlife.com/wiki/Listen>
- [26] Second Life Wiki (2009). LSL Wiki: lIRezObject. Linden Research, Inc. last retrieved Aug. 19th, 2010, from <http://wiki.secondlife.com/wiki/LIRezObject>
- [27] SimTeach (2010). Second Life Education Wiki. SimTeach, last modified Feb. 2nd, 2010; last retrieved Feb. 12th, 2010, from http://www.simteach.com/wiki/index.php?title=Second_Life_Education_Wiki
- [28] Smith, B. L., & MacGregor, J. T. (1992). What is collaborative learning? In Goodsell, A. S., Maher, M. R., and Tinto, V. (Eds.), Collaborative Learning: A Sourcebook for Higher Education. National Center on Postsecondary Teaching, Learning, & Assessment, Syracuse University.
- [29] Stacey, E. (1999). Collaborative Learning in an Online Environment. The Journal of Distance Education, VOL. 14, No. 2, 14-33.
- [30] Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences (pp. 409–426). Cambridge, UK: Cambridge University Press.
- [31] Wikipedia (2010c), Google Web Toolkit, Wikipedia, the free encyclopedia, last edited Aug. 18th 2010, last retrieved Aug. 18th 2010 from http://en.wikipedia.org/wiki/Google_Web_Toolkit
- [32] Willkomm, S. (2009). The virtual presence of the TU Graz in Second-Life. Master thesis, Graz University of Technology, Graz, Austria, 2010.

Acknowledgements

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

Authors:

Christian Gütl Dipl.-Ing. Dr. techn.
Graz University of Technology, Institute for Information Systems and Computer Media
Inffeldgasse 16c, A-8010 Graz
School of Information Systems, Curtin University of Technology
G P O Box U1987, Perth, Western Australia, Australia
cguetl@iicm.edu

Vanessa Chang Ph.D.
School of Information Systems, Curtin University of Technology
G P O Box U1987, Perth, Western Australia, Australia
vanessa.chang@curtin.edu.au

Stefan Freudenthaler, Master Student
Graz University of Technology, Institute for Information Systems and Computer Media
Inffeldgasse 16c, A-8010 Graz
School of Information Systems, Curtin University of Technology
G P O Box U1987, Perth, Western Australia, Australia
stefan.freudenthaler@student.tugraz.at