Collaborative Virtual 3D Environment for Internet-accessible Physics Experiments

Tina Scheucher[1], Philip H. Bailey[2], Christian Gütl[1,3], V. Judson Harward[2]

[1] Graz University of Technology, Graz, Austria

[2] Massachusetts Institute of Technology, Cambridge, USA

[3] Curtin University of Technology, Perth, WA

Abstract—**Immersive 3D worlds have increasingly raised the interest of researchers and practitioners for various learning and training settings over the last decade. These virtual worlds can provide multiple communication channels between users and improve presence and awareness in the learning process. Consequently virtual 3D environments facilitate collaborative learning and training scenarios.**

In this paper we focus on the integration of internet-accessible physics experiments (iLabs) combined with the TEALsim 3D simulation toolkit in Project Wonderland, Sun's toolkit for creating collaborative 3D virtual worlds. Within such a collaborative environment these tools provide the opportunity for teachers and students to work together as avatars as they control actual equipment, visualize physical phenomenon generated by the experiment, and discuss the results. In particular we will outline the steps of integration, future goals, as well as the value of a collaboration space in Wonderland's virtual world.

Index Terms—**Collaborative immersive environment, virtual worlds, Project Wonderland, iLabs, TEALsim, physics experiments**

I. Introduction

The use of 3D virtual learning environments in education has grown considerably over the past years and has captured the attention and interest of educators worldwide. This interest originally arose from educators' demands for more flexible and expressive technologies to support the students' learning experiences. Despite rapid development, many educational application domains continue to exhibit a great need for improvement. If we focus on physics education from secondary school through university, the main pedagogical problem remains that students have great difficulty in relating physics theory and equations to the phenomena they observe in the lab or, indeed, in the everyday world. Perhaps the most promising tool for improving students' ability to relate physics theory to observable phenomena is the use of 3D visualization.

A. 3D Virtual Environments

3D virtual environments (VEs) make use of the third dimension to increase the participant's sense of reality. They consist entirely of 3D components, with which the user is able to interact within the environment. For example, the user can enter and exit rooms, walk around buildings, and open drawers to see what is inside. Increases in desktop 3D computer graphics and network infrastructure were the two main technological advances that have enabled the development of 3D VEs and have generally increased the potential of the World Wide Web. Such environments provide the illusion of being immersed within a 3D space, and enable the user to perform actions and behaviors which are analogous to those she can initiate in the real world [16]. The fact that users can gain experience in the same way that they can in the real world opens new and interesting opportunities for physics education. By expanding reality beyond the limits of the classroom in both time and space, the educator can enrich the student's learning experience to ease understanding of abstract physics concepts [8].

3D VEs designed to improve learning processes and enrich learning resources are called 3D virtual learning environments (VLE's) [17]. Rather than attempting to recreate common face-to-face classroom settings, the aim of these environments is to complement and expand classroom activities. In addition to providing more flexible learning resources than those available in conventional 2D e-learning environments, VLEs also try to use the third dimension to enhance collaborative learning and provide hands-on learning experiences not possible in the classroom. Through facilitating 3D visualizations, these environments have the ability to make the unseen seen. A 3D simulation generated from a formal mathematical model embodying the laws of physics becomes a controllable computer-generated analog of real-world objects or processes, which the student can use to understand the behavior of complex systems [2] [19] [4]. Along with visualizing physics phenomena and simulating their actions/behavior, a 3D VLE also provides standard, quasi-realistic mechanisms for the user to interact with the underlying physical/mathematical models. Computer-generated simulations can enable the students to compare theoretical with experimental results, since they embody learning by doing [6].

B. Collaborative Learning Environments

While several VLE's for virtual physics experiments are designed as single-user environments, wherein learners accomplish experiments by themselves, recent approaches demonstrate that sharing experiences in a group, discussing results together, and interacting with other participants can bring learners to a higher achievement level [14]. Multi-user VLEs, also known as collaborative VLEs (CVLE) [18] or virtual worlds [1], enable interactions not only with objects but also with other users. These virtual worlds are richly immersive, and learners participate as avatars, which represent the learners within the virtual space. Multiple learners can explore the same 3D collaborative environment, can communicate with other learners, and can move their avatars through the space as if they were physically walking [5]. CLVEs have achieved the goal of virtual collaboration, therefore they have the potential to offer important influences on the 3D immersive virtual world approach [3] [15].

The goal of our project is to provide a 3D CVLE for physics education in which students and educators are able to work together in a collaborative way. Through the integration of specific technologies and projects, the users of this immersive environment use a 3D simulation to visualize the real behavior of an actual remote physics experiment. The following section of this paper will introduce the software and technologies which are used to realize this collaborative learning space. We will describe both the current implementation as well as our planned future work.

II. Background Technologies

Our research is based on multiple pre-existing projects which embody virtual technologies. They each have their respective benefits, and our goal has been to fuse them into a new collaborative learning environment.

A. *Project Wonderland*

Project Wonderland [25] is a 100% Java open source toolkit for creating collaborative 3D virtual worlds, developed by Sun Microsystems Inc [21]. This multi-user immersive environment incorporates 3D and desktop computer graphics technology with digital media, to immerse and engage users in the same way 3D games grab and keep the attention of players. Wonderland offers students a virtual world within which to work and learn together without the need for a separate collaboration tool [22]. Wonderland focuses on providing open source tools for collaboration. These tools include high quality spatially localized audio chat, shared whiteboards, HTML and pdf viewers, streaming video, and shared virtual desktops. Because the system is open-source, it can be extended by creating new tools.

Wonderland release 0.4 is based on a client/server architecture using the Sun Game Server, Project Darkstar [23]. Client/server architectures are well suited for game systems since the server can coordinate the state of all client instances.

Besides the Darkstar server, Project Wonderland relies on further open-source projects developed by Sun Microsystems, including Java3D [10], Project Looking Glass [24] and jVoiceBridge [12]. Java3D, a scene graph-based 3D application programming interface (API) for the Java platform, represents and renders the scene graph on which the 3D world is built. Project Looking Glass is built upon Java3D and provides the 3D scene manager. Finally jVoiceBridge provides communication in the virtual world via real-time immersive stereo audio with distance attenuation.

B. *The iLab Project*

The iLab Project [27] developed at MIT provides an infrastructure for the design and sharing of Internet accessible lab experiments. It allows students and educators remote access to actual equipment in real laboratories. Unlike conventional experiment facilities, the equipment is available 24/7, from anywhere at any time, and may be easily shared. Online laboratories offer the possibility to expand the range of experiments that students use in the course of their education. The iLab Project's vision is to share lab experiments with other institutions across the world.

C. *The TEAL Simulation System (TEALsim)*

TEALsim [26] is an open source Java3D based simulation toolkit, developed at the Massachusetts Institute of Technology (MIT) as a component of the 'Studio Physics' [20] Technology Enabled Active Learning Project (TEAL) [26]. TEAL's major goal is to help students conceptualize phenomena and processes especially in electromagnetism [7]. The approach of this project is to provide a combination of lectures, lab experiments and a range of simulations in a specially designed classroom to facilitate group interaction. Students typically work in groups of three.

TEALsim simulations are already effectively integrated in the TEAL curriculum. They are used to illustrate abstract physics concepts visualized by means of a 3D model. To give one example, electromagnetic fields and phenomena are typically not visible in real settings, but using such a simulation engine, students can view a synchronized visualization that renders an accurate representation of the invisible field lines. Student lab groups have the ability to interact with the model using their desktop input devices, control input values, and then see how the simulation is effected by their changes. In addition, 3D simulations can be carried out as often as needed to give the students a better understanding for physics concepts.

D. *The Force On A Dipole Experiment*

The "Force on a Dipole" experiment developed at MIT combines an iLab interactive lab and a TEALsim visualization. The experiment itself consists of a small magnet suspended vertically by a spring in the center of two horizontally mounted coils. An electrical current may be directed through the top or, through both coils. A video camera is stationed to observe the position of the magnet (Fig.1).

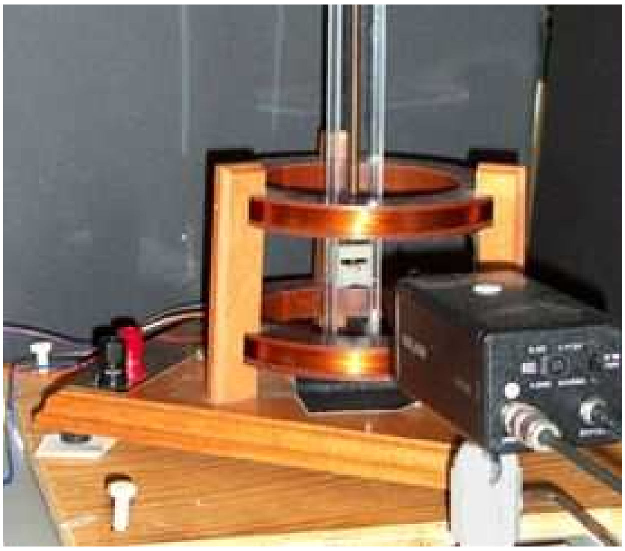

Figure 1. Force on a Dipole experiment located at MIT

This experiment is controlled by a LabVIEW [13] application (Fig. 2), that allows users to configure the current amplitude and frequency of the current as well as to specify to which coils it is directed. The application displays the input current, the current in the top coil and the position of the magnet. In addition, users may request that data be collected, and a remote process may request that the experiment data be streamed to it using TCP/IP. Remote access to this experiment is provided by an iLab LabServer.

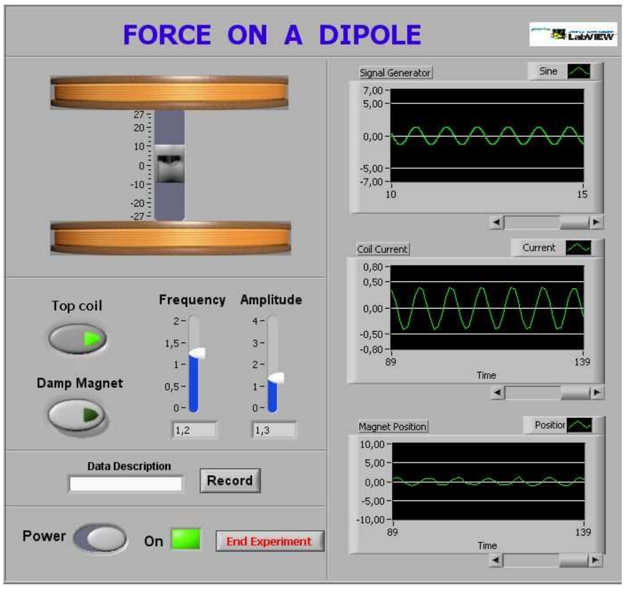

Figure 2: LabVIEW application Front Panel

Using the TEALsim Toolkit, a specialized application [9] was built to subscribe to the experiment data stream and to use the data to drive a real-time visualization of the experiment on the student's computer. The visualization shows what is physically happening in the experiment, as well as "augmenting reality" with representations of the magnetic and/or electric field lines. This implies that the LabVIEW interface enables remote access to control the experiment while the TEALsim software creates a synchronized 3D simulation of the magnetic field lines to illustrate the interaction of the stationary current carrying coil with the suspended permanent magnet (Fig. 3).

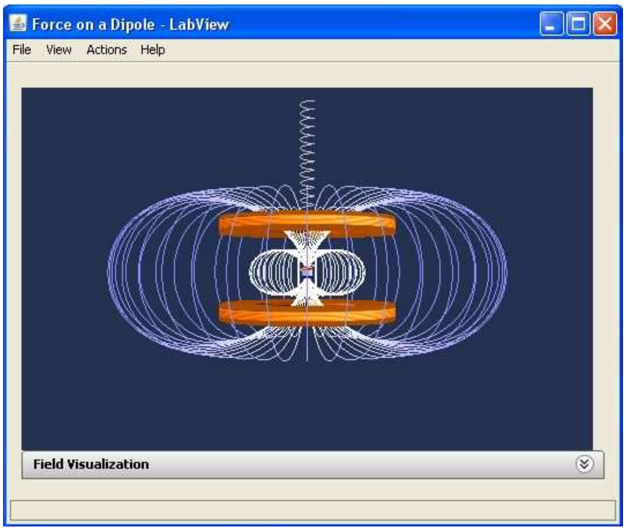

Figure 3. TEALSim visualization of Force on a Dipole experiment

The combination of these projects provides a powerful tool to enhance understanding of abstract physics concepts using active and passive visualizations, especially in areas where we can make the "unseen seen", as in electromagnetism. These educational benefits are now available to students and educators around the world.

In trying to extend the established benefits of the present “Force on a Dipole” project, the current team conceived the idea of a collaborative immersive environment for physics education. The TEAL Studio Project already reflects this approach by providing a physical collaborative learning environment. But it assumes that students are physically collocated and can physically interact. The current project aims to achieve the same goal when that is not possible. In the next section, we describe the realization of this collaborative environment using Project Wonderland tools and the integration of the “Force on a Dipole” project.

III. Collaborative Virtual Learning Environment

We will first explain the basics of Wonderland's infrastructure in order to give a better understanding of how visualization and interaction work in Wonderland.

At the time we started with our research work, Project Wonderland's release 0.4 was the stable version to create a 3D world space. This is the version on which the current project is based although we expect to migrate it to future releases of the Wonderland environment.

A. *Wonderland's Cell Architecture*

Wonderland organizes the 3D virtual world in cells. A cell represents a volume of space in the world, such as part of a scene or an object. Avatars are also represented through cells. The cell is composed of both a client and a server object. The server part, instantiated by the game server Project Darkstar, consists of a single class that maintains the shared state of the cell across all client participants. The client-side part is responsible for rendering the cell in-world either by loading an art resource or by drawing objects directly using the Java3D APIs. Cells divide the world's Java3D scene graph into networked-optimized chunks. For instance, the WorldRoot Cell, which represents the whole world, may consist of several Room Cells, and each of these Room Cells may consist of an Application Cell, Avatar Cell, etc. Two important scene graph properties of a client-cell are the world's scene graph node and the cell's scene graph node.

Communication between client and server is realized through client-server messages. If a client action causes a client-cell process state change, a message will be sent from client to server to update the server-side instance of this cell. The server cell is required to send messages to all of its other client cells in the world to update their instance of the cell.

The absolute 3D-coordinates of a cell's origin within the world as well as the name of the class to which the cell object belongs is provided in an XML-formatted cell descriptor file within the Wonderland-File-System (WFS). The WFS records instances of cells that are added to a world. When the Project Wonderland-Server is started or reloaded, the WFS is read and an instance of a cell is created on the server for each XML cell descriptor in the WFS. When a client connects to the server the client-side cells are created for each cell currently in the server.

B. *Game-server infrastructure*

Project Wonderland is also based on the game server Project Darkstar, which provides the infrastructure to implement the virtual world application. The major components of this infrastructure include a transaction model and a network protocol. The transaction model manages access to the Darkstar server. Thus, if multiple clients want to perform an action and update the server at the same time, the server has to synchronize the client updates so they do not interfere with each other. If any modification happens at a client's cell, the server-side piece of the cell is updated in a thread-safe manner. Then the server has to communicate the change to all the other clients in the world via a message send. The network protocol assures that the message is received by the recepient.

C. *InterCell Communication*

The message mechanism provided in Wonderland 0.4 is designed in such a way that the Wonderland server-side cell automatically sends a message to each client-side cell to trigger the updating of the client's state. What happens if the state of another cell is supposed to change? That means, if an update of another special cell is desired, a communication channel between different cells is required. Since such Inter-Cell communication is not provided by the Wonderland 0.4 toolkit, it was part of our research work to support this kind of communication. See starting the simulation, described in section G.

D. *ForceOnADipole Cell*

In our project, integrating the TEAL simulation of the Force on a Dipole experiment required the creation of a new cell type with server-side and client-side classes. The server-side cell TEALSimCellGLO has to communicate with the client-side TEALSimCell when the client starts in order to setup the cell's properties and to update them during the application runtime. The major tasks of the client-side TEALSimCell, besides providing access to the cell's scene graph, is to notify the server of the user interactions, and to process the setup and update messages from the server-side cell.

E. *Integration of the ForceOnADipole Model*

TEALsim simulations are Java classes that have a well defined interface (TSimulation) and consist of three major components, the simulation engine, the rendering engine, and the user interface. When running a simulation on the desktop or as an applet, the simulation is run within a TEAL framework which is responsible for loading the simulation and presenting the user interface. TSimulations were not designed to run outside of a framework, for this project the TEALSimCell is used as the framework.

The TEALsim render engine is implemented as a Java3D-based viewer that manages the simulation's scene graph. Project Wonderland's 3D components are also based on Java3D. This fact made it possible to integrate the two technologies with a few changes to the TEALsim architecture.

When the Wonderland client is started, the TEALSimCell creates an instance of the ForceOnADipole simulation. Through this instance the cell gets a pointer to the root of the ForceOnADipole simulations's scene graph. This scene graph has to be integrated into the entire scene graph of Wonderland by attaching it to the cell's scene graph node. Since the TEALsim scene graph was attached to the TEALsim viewer, it has to be 'detached' before this can be done. Finally this entire subgraph is then said to be live in Wonderland's world.

The simulation engine maintains a model of the current state of all simulated objects and constantly integrates the interaction of all objects in the simulation. Once the next state of the model is resolved, the render engine is notified and the scene graph is updated. For performance reasons the simulation engine runs within its own thread.

To control the appearance of the 3D simulation inside Wonderland, the actual experiment must be started before the simulation can connect to the data stream.

F. *Connection to the actual Experiment*

The iLabs framework provides access to the remote iLab LabServer, that launches the interactive LabVIEW application. This application provides access to the actual hardware executing the experiment and to a GUI front panel which controls the output and input parameters of the experiment.

In order to start and control the actual experiment in world, the GUI front panel of the LabVIEW application needs to be displayed. For this purpose, Wonderland supports a graphical desktop sharing system, called Virtual Network Computing (VNC), to control another computer remotely. A VNC viewer, integrated as a cell in Wonderland, can connect to a VNC server on the same or another computer. The VNC server, in our context, is located on the same server where the LabVIEW software is running. In the world a VNC viewer is used to display the GUI front panel of the LabVIEW application. User interaction is provided through keyboard or mouse events, which are transmitted from the computer on which the Wonderland client is running to the computer where the LabVIEW application is hosted. Multiple clients may connect to a VNC server at the same time. Therefore, avatars have the ability to launch and interact with the real experiment via the VNC viewer just as students in the classroom version of TEAL can control the frequency and amplitude of the current.

G. Starting the simulation in world

After the real experiment is started, avatars can now start the virtual representation of the experiment in world. In the TEAL framework, this action is triggered by selecting a menu item, but within Wonderland 0.4 the simulation's user interface components are not available. The learning space must provide a button, implemented as an Inter-Cell communication cell. When the button is pressed, a message is sent to the button's server-side cell, which then sends a message to the server-side TEALSimCellGLO. A message is sent from the server-side cell to each of the client cells requesting that the client's simulation model connect to the LabVIEW Application, each simulation opens a data socket to the LabVIEW application.

The simulation engine thread of the TEALsim simulation gets data via the socket channel and updates simulation objects to reflect these new values. It also informs the viewer's rendering engine of any visual changes to the simulation. To reflect the effect of the user's input on the simulation in Wonderland, the world's scene graph must be updated. After starting the socket, we connect again to the actual branch group of the rendering engine and append it to the world's scene graph. In order to update the client itself and all other clients to reflect the user input in the simulation, the TEALSimCell sends a message to the TEALSimCellGLO. This server side implementation has the important task of relaying the message to all clients. When the client-side representation of the start button is initially pushed, the server cell must start the connection on all other client cells. Conversely, if the button is pressed again, clients have to be informed to stop the socket in order to end the simulation's connection. To represent application state, the button will turn to red when it's pushed to start the simulation, and to green after pushed to stop the simulation.

H. Real-time videostreaming via webcam

For the ForceOnADipole application there is also a webcam provided, which is not part of the application but access to the webcam is facilitated by the iLab ServiceBroker. Wonderland also allows students to stream the actions of the real hardware, realized in a separate cell. This cell facilitates a video screen in world where avatars are able to see the working experiment via the network camera. The specified camera in the 'video-webcam-wlc.xml' is the AXIS212PTZCamera, but it's possible to add support for other Axis network cameras, like the Axis 211 we are using. The video is streamed to all clients and played concurrently in the browser of all avatars in the same world, as long as the actual experiment is still connected. Control buttons allow the avatars to terminate and control the video.

I. Communication and Collaboration tools in world

Wonderland supports student-to-student, student-to-educator or educator-to-educator communication via text chat as well as audio chat. These communication tools are important for discourse-based collaborative learning.

Wonderland also incorporates collaboration tools to support higher level representation and participant co-awareness in coordinated learning activities. In certain cases there is a need in education to explain things in a different way, like a chalkboard in most classrooms. The chalkboard is one of the most basic forms of instructional media and is best used for emphasizing essential information and developing ideas as the class progresses. To facilitate this important subject also in a virtual learning environment, Wonderland provides a shared electronic whiteboard.

IV. Current Status and Future Work

Initially, we intended to implement on top of the newer version 0.5 of Project Wonderland, because of major architectural and graphics quality improvements, although this would preclude the use of the same Java3D scene graph for both Wonderland and TEALsim. But Wonderland's new version 0.5 is being issued as a series of monthly developer releases which began in December 2008, with the expected end-user release due sometime during the summer 2009. Since our research work started at the end of January, 2009, and the timeline of our first author required us to develop the first version of the collaborative learning environment before the end of July, we decided to use version 0.4 for this project. In effect, when we started, version 0.4 was both more functional and more stable.

Starting work on version 0.4 was the right decision, because we were able to develop our proof-of-concept virtual learning environment in Wonderland within a short period. We were able to determine if there are some performance issues in running TEAL simulations in Wonderland. But the simulation of the force on a dipole experiment proved to work well, although there is certainly some room for improvement, especially when many avatars are present at the same time.

Despite the partial lack of documentation on building a 3D environment in Wonderland, we were able to make good progress with help from the Wonderland developers' team. The current state of our work is a working collaborative virtual environment in Wonderland 0.4, which currently provides nearly the full functionality of the force on a dipole experiment.

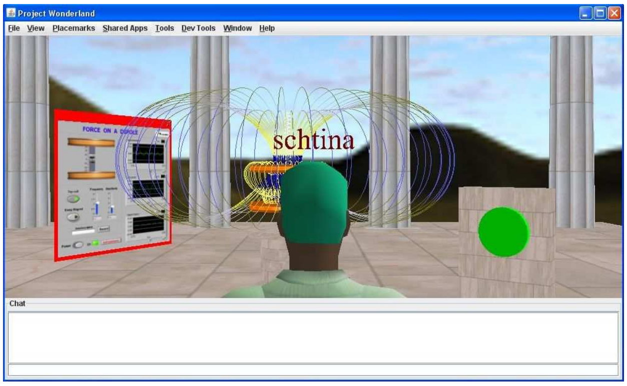
Project Wonderland
File View Placemarks Shared Apps Tools Dev Tools Window Help
FORCE ON A DIPOLE
schtina
Chat

Figure 4. TEALsim visualization of 'Force on a Dipole' in Wonderland

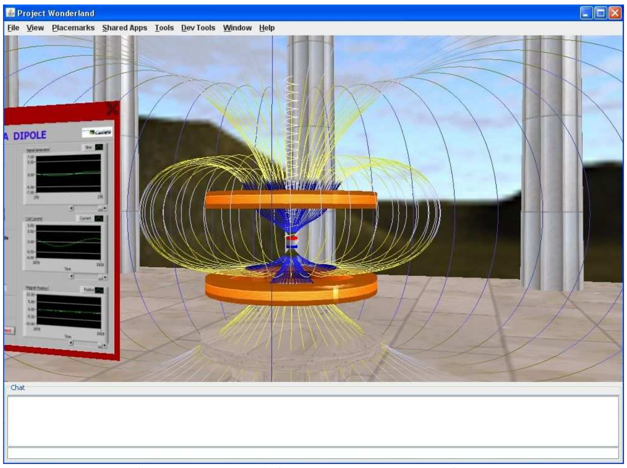

Figure 5. Close-up view of the 'Force on a Dipole' Experiment in Wonderland

For the future, the plan for this research project is to port from 0.4 to 0.5 to achieve a richer 3D experience within the world.

Wonderland is moving from a Java3D rendering engine to JMonkeyEngine (JME) [11], which should cure the limitations imposed on Wonderland by the visual and design constraints of Java3D. JME is a full featured 3D game engine which is already in use by professional game companies. JME should provide higher performance as well as better graphics capabilities. This will require replacing the Java3D rendering engine used by TEALsim and a redesign of the TEALsim 3D viewer.

A further important change in Wonderland 0.5 will replace the existing inflexible avatar system with a modern state-of-the-art avatar system including new bone models and advanced skinning features, which will bring the appearance of an avatar to a higher level.

Finally, version 0.5 will provide embedded swing functionality for general user interface construction. With this feature we get the ability to create our own Swing-based user interface to interact with the experiment instead of using a VNC-viewer to show the LabVIEW application. Our goal is to support the TEAL simulation's user interface components with this Swing feature.

After porting to 0.5, the next step in the future will be extending the learning space by populating the virtual learning environment with more simulations.

V. CONCLUSION

In this paper, we have set out our research work and vision for collaborative, simulation-based learning in desktop 3D virtual environments. The arrival of the desktop variant of 3D virtual environments has made the technology widely accessible and has dramatically reduced the costs of the necessary hardware.

The obvious advantage of virtual laboratories lies in reducing the number of expensive physical laboratories that must be maintained and in allowing students remote access at any time. Support for teaching laboratories is significant, and so there is a real potential for allowing widespread use of virtual laboratories. The use of such laboratories in a collaborative virtual learning environment improves the potential effect of virtual laboratories by allowing collaboration between students and teachers and between students themselves. It also enables global collaboration between students by forming learning teams across continents in order to solve problems.

Besides providing a technical solution to realize a collaborative learning environment, we believe that it is crucial to establish a new learning culture with horizontal communication structures. Active participation and meaningful discussion are the collaborative values we want to support in our virtual environment.

ACKNOWLEDGMENT

We would like to thank MIT's Center for Educational Computing Initiatives for hosting this research project. We would like to thank the staff of the Wonderland developer's team for their knowledgeable support.

REFERENCES

[1] R.A. Bartle, "Designing Virtual Worlds", New Riders Publishing, Indianapolis, 2004, Chapter 1, pp. 2.

[2] Bouras, C., & Tsiatsos, T. (2006). Educational virtual environments: design rationale and architecture. Multimedia Tools and Applications, 29, 2, 153–173.

[3] S. Bronack, R. Sanders, A. Cheney, R. Riedl, J. Tashner, & N. Matzen, "Presence pedagogy: Teaching and learning in a 3D immersive world". International Journal of Teaching and Learning in Higher Education, 2008, 20(1), pp. 59-69. Retrieved May 8th, 2009 from: http://www.isetl.org/ijtlhe/pdf/IJTLHE453.pdf

[4] Y.S Chee, C.M. Hooi, "C-Visions: Socialized Learning through Collaborative, Virtual, Interactive Simulations". Published in Proceedings of CSCL 2002: Conference on Computer Support for Collaborative Learning, Boulder, CO, USA, pp. 687–696., Retrieved 10th May, 2009 from:http://yamsanchee.myplace.nie.edu.sg/Publications/2002/CSCL2002CheeHooi.pdf

[5] Cherney et al., "Teaching and Learning in a 3D Immersive World: The AETZone Model", 2007. Retrieved May 16th, 2009 from http://download.101com.com/CAM/conf/2007/W02.pdf

[6] B. Dalgarno, "The Potential of 3D Virtual Learning Environments: A Constructivist Analysis", *Electronic Journal of Instructional Science and Technology, 5(2), pp. 3-6, 2002.* Retrieved May 8th, 2009 from: http://www.usq.edu.au/electpub/e-jist/docs/Vol5_No2/Dalgarno%20-%20Final.pdf

[7] Y.D. Dori, J. Belcher, "How does technology-enabled active learning affect undergraduate students' understanding of electromagnetism concepts?" The Journal of the Learning Sciences, 14(2), pp. 243-279., 2005. Retrieved May 10th, 2009 from:http://edu.technion.ac.il/chemical-education/judy/publications/no9_TEAL%20JLS%202005%20Dori%20&%20Belcher.pdf

[8] D. Economou, W. L. Mitchell, T. Boyle, "Requirements elicitation for virtual actors in collaborative learning environments", Computers & Education 34, pp. 225-239, 2000.

[9] ForceOnaDipole, Retrieved May 20th, 2009 from https://wikis.mit.edu/confluence/display/ILAB2/teachspin_ilab

[10] Java3D, Retrieved May 20th, 2009 from http://java.sun.com/javase/technologies/desktop/java3d/

[11] jMonkeyEngine, Retrieved May 20th, 2009 from http://www.jmonkeyengine.com

[12] jVoiceBridge, Retrieved May 20th, 2009 from https://jvoicebridge.dev.java.net/

[13] LabVIEW, Retrieved May 20th, 2009 from http://discoverlab.com/References/WP2238.pdf

[14] J. LAISTER, S. KOBER, "Social aspects of Collaborative Learning in Virtual Learning Environments". Proceedings of the Networked Learning Conference Sheffield, March 2002, Retrieved May 15th, 2009 from http://comma.doc.ic.ac.uk/inverse/papers/patras/19.htm

[15] M.J.W. Lee, "How can 3D virtual worlds be used to support collaborative Learning?", An Analysis of cases from the literature. Retrieved May 10th, 2009 from http://www.je-lks.it/en/09_01/3met_lee_ing09.pdf

[16] M. Mine, “Virtual Environment Interaction Techniques”, UNC Chapel Hill, Computer Science Technical Report, TR95-018, 1995. Retrieved May 20th, 2009 from http://www.macs.hw.ac.uk/modules/F24VS2/Resources/VEinteractionTechniques.pdf

[17] D. Nonis, “3D virtual learning environments (3D VLE)”, Ministry of Education, Singapore, 2005. Retrieved May 8th, 2009 from: http://www.moe.gov.sg/edumall/rd/litreview/3d_vle.pdf

[18] B.L. Smith., J.T. MacGregor, "What is Collaborative Learning?" In A.S. Goodsell, M.R. Maher, and V. Tinto (eds.), Collaborative Learning: A Sourcebook for Higher Education. University Park, 1992, Retrieved May 16th, 2009 from http://learningcommons.evergreen.edu/pdf/collab.pdf

[19] R. Smith, “Simulation: The engine behind the virtual world”. In the *Simulation 2000 series, Volume 1, Retrieved May 19th, 2009 from*http://www.modelbenders.com/papers/sim2000/SimulationEngine.PDF

[20] Studio Physics, Retrieved May 20th, 2009 from http://www.rpi.edu/dept/phys/research/educ_physics.html

[21] Sun Microsystems, Inc, Retrieved May 20th, 2009 from http://www.sun.com

[22] Sun Microsystems, Inc., “Immersive Education with Project Wonderland”, Groundbreaking Virtual-Reality learning systems, Retrieved May 20th, 2009 from
http://www.sun.com/solutions/documents/solution-sheets/ed_immersive_education.pdf

[23] Sun Microsystems, Inc., Project Darkstar. Retrieved May 20th, 2009 from http://projectdarkstar.com/,

[24] Sun Microsystems, Inc., Project Looking Glass. Retrieved May 20th, 2009 from http://www.sun.com/software/looking_glass

[25] Sun Microsystems, Inc., Project Wonderland: Toolkit for Building 3D Virtual Worlds. Retrieved May 20th, 2009 from https://lg3d-wonderland.dev.java.net

[26] Technology Enabled Active Learning (TEAL), Visualizing Electricity and Magnetism at MIT, Retrieved May 20th, 2009 from http://web.mit.edu/8.02t/www/802TEAL3D/teal_tour.htm

[27] The iLab Project, Retrieved May 20th, 2009 from https://wikis.mit.edu/confluence/display/ILAB2/Home

AUTHORS

Tina Scheucher is with the Technical University of Graz, Graz, 8010 Austria (e-mail: schtina@sbox.tugraz.at)

Philip H. Bailey is with the Massachuesetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: pbailey@mit.edu).

Christian Gütl is with the Technical University of Graz, Graz, 8010 Austria and the Curtin University of Technology, Perth, Australia (e-mail: cguetl@iicm.edu).

Judson V. Harward is with the Massachuesetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jud@mit.edu).

Manuscript received 3 June 2009.

This work was supported in part by the Austrian Marshall Plan foundation

.

Published as submitted by the author(s).