Use of Open Source Software and Virtualization in Academia to Enhance Higher Education Everywhere

Maurice E Dawson, Jr.
Imad Al Saeed
USE OF OPEN SOURCE SOFTWARE AND VIRTUALIZATION IN ACADEMIA TO ENHANCE HIGHER EDUCATION EVERYWHERE

Maurice Eugene Dawson Jr. and Imad Al Saeed

ABSTRACT

As costs around the world continue to rise for education, institutions must become innovative in the ways they teach and grow students. To do this effectively, professors and administrative staff should push toward the utilization of Open Source Software (OSS) and virtual tools to enhance or supplement currently available tools. In developing countries, OSS applications would allow students the ability to learn critical technological skills for success at small fraction of the cost. OSS also provides faculty members the ability to dissect source code and prepare students for low-level software development. It is critical that all institutions look at alternatives in providing training and delivering educational material regardless of limitations going forward as the world continues to be more global due to the increased use of technologies everywhere. Doing this could provide a means of shortening the education gap in many countries. Through reviewing the available technology, possible
implementations of these technologies, and the application of these items in graduate coursework could provide a starting point in integrating these tools into academia. When administrators or faculty debate the possibilities of OSS, gaming, and simulation tools, this applied research provides a guide for changing the ability to develop students that will be competitive on a global level.

**INTRODUCTION**

This book chapter will cover the utilization of virtualization, Open Source Software (OSS), and simulation tools that are also OSS. OSS can be defined as software that is made available in source code form. This is important as this source code may fall under the General Public License (GPL) which is a widely used free software license that is managed under the GNU Not Linux (GNU) Project (GNU, 2007). Virtualization is important as this is an effective method to reproduce system learning environments as a virtual instance reducing the overall hardware footprint and need to for a massive lab. This chapter will also cover two simulation tools to include correlating published research on how these items were integrated into graduate systems engineering coursework.

**VIRTUALIZATION**

In terms of virtualization, there are available tools to create a virtual version of a system. In terms of educational resources this provides a method for institutions to train on virtual machines (VMs). This allows a university to teach students complex techniques to computer science, engineering, or information technology (IT) such as networking, programming, system administration, and Information Assurance (IA). There are multiple types of virtualization such as hardware, desktop, memory, storage, data, and network. In Fig. 1 displayed is a screenshot of Ubuntu 11.10 running in a VM on the Windows 7 desktop.

For institutions that would like the opportunity to provide a cloud-like environment, tools such as Oracle Virtual Box and Vmware Player provide that ability. However, it should be noted that new Linux distributions running that require GNOME 3 will have issues running on older hardware. With older hardware as a constrain, there are bare minimal Linux
distributions such as Puppy Linux and Damn Small Linux (DSL). VMs provide the ability for a student to experiment with hundreds of Operating Systems (OSs) without installing or uninstalling the base OS.

Additionally, this allows for the creation of baseline OS images for classes. For example, a marketing course would have an OS created with all the software, case studies, etc. preloaded. This baseline OS for marketing would have statistics software, graphic design software, social marketing tools, case studies, eBooks, links to online course management tool, etc. This would allow an institution to have an image ready for every class to ensure consistency, and that the students have all required tools needed. In the case for a more technical course such as software engineering, the students would have a baseline OS image with all the programming software, the integrated development environment (IDE), quality testing tools, etc. preloaded.

For professors, this virtualized environment would allow for the monitoring, distribution, and quicker deployment of available tools. This environment would be a cloud computing solution. Cloud computing is based on concepts of virtualization, distributed computing, and networking and is underpinned in the latest web and software technologies (Vouk, 2008). A useful definition of cloud computing is that it is a way of delivering applications as services over the Internet as well as a way of providing for the hardware and system software that act as platforms for these

---

**Fig. 1.** Screenshot of Ubuntu VMware VM Running on Windows 7 Desktop.
applications and services (Armbrust et al., 2009). Cloud is also used to refer to a network of computers that are linked together and distribute processing capacity and applications to different systems (Johnson, Levine, & Smith, 2009). Cloud computing lets organizations add on to their IT and computing capacity without having to invest in new architecture, software, or hardware or in training and developing personnel (Glotzbach, Mordkovich, & Radwan, 2008). A cloud environment could prove to be a cost-effective implementation of which would allow for scalability if these right tools are utilized.

**ADDING AND EXPANDING CAPABILITIES WITH LINUX**

In terms of virtualization, the most cost-effective method is with the use of Linux as the OS. As institutions around the world look to provide their students and faculty with the ability to work in highly technical or large demand fields, it is imperative that all institutions have the ability to provide a simulated environment to teach the necessary concepts such as program management, design, and engineering. However, to do this at a fraction of the cost, Linux and other OSS are vital for implementation.

*Why We Need to Consider Linux Essential in Higher Education*

Linux is a Unix like OS that is built on the Linux kernel developed by Linus Torvalds with thousands of software engineers. As of 2012, there are over 200 active Linux distributions. The majority of the kernel and associated packages are free and OSS. This type of software provides a license that allows users the right to use, copy, study, change, and improve the software as the source code is made available. Providing source code allows developers or engineers to understand the inner workings of development. Imagine being able to study Mac or Windows by viewing all the source code to replicate similar developments. This exercise would be great for a developer to learn low-level coding techniques, design, integration, and implementation.

In terms of associated cost the majority of Linux distributions are free. However, some distributions require a cost for updates or assistance that related to specific needs such as OS modifications for server hosting. In software, there is a packet management system that automates the process
of installing, configuring, upgrading, and removing software packages from an OS. In the Linux OS builds, the most common packet management systems are Debian, Red Hat Package Manager (RPM), Knoppix, and netpkg. Below are a list of some Linux distribution and potential uses in education (Table 1).

Other countries are supporting the OSS movement as well. In China, Red Flag Linux commands over 30 percent of the market (Pan & Bonk, 2007). China is actively looking for an OS to combat Windows OS, thus the momentum for OSS continues to grow. In Russia, Linux may become a national OS by 2015 as they are as well looking for lower cost solutions in all levels of education. The Edubuntu OS, which has roots in South Africa, is being utilized by the Republic of Macedonia in all K-12 schools. With software packages such as LibreOffice students and faculty have the ability to perform similar functions as those found in the Microsoft Office suite without having to spend any money to obtain the software (Fig. 2).

Since the early 2000, there has been significant encouragement for the use of Linux in the Spanish public school system (Munoz et al., 2012). This use has been at the K12 level; however, the university level has yet to fully integrate this technology into the classroom. However, in a survey conducted by Accenture over 300 large blue chip organizations utilize OSS (Accenture, 2010). This indicates that there is a growing need for organizations to have employees familiar with OSS tools for development to include those for management.

**OSS to Assist with the Development of Project Management Concepts and Tools**

The need and presence of project management knowledge has developed to become a necessity in many organizational industries, while project management methods, processes, and certifications have become accepted standards in many industries (PMI, 2010). In 2008, Global Knowledge and Fortune Magazine listed project management in their top 10 career of choice, while in a recent salary survey conducted by ZDNET’s Tech Republic organization, the PMP (Project Management Professional) certification was listed as the highest paying certification to have in the technology industry, while other project management certifications governed by the International Project Management Association (CPD, CPM) and Office of Government Commerce (PRINCE2) are highly sought after in European industries.
<table>
<thead>
<tr>
<th>Linux Distributions</th>
<th>Description and Potential Use</th>
<th>Packet Management System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu</td>
<td>One of the most popular Linux OS developed to be a complete OS that can be an easily replaced for other comparable OSs</td>
<td>Debian-based</td>
</tr>
<tr>
<td>Edubuntu</td>
<td>OS targeted for grades K12. Contained in OS are tons of software applications that are useful to those who are education majors</td>
<td>Debian-based</td>
</tr>
<tr>
<td>Damn Small Linux</td>
<td>This OS is designed as a small OS to be utilized on older hardware. This OS is great for institutions that have old computers and want to revitalize them for use. OS is also great for VMs as DSL requires a low amount of memory</td>
<td>Knoppix-based</td>
</tr>
<tr>
<td>BackTrack</td>
<td>OS based on Ubuntu for digital forensics and penetration testing. Great tool for students majoring in technology fields. As cyber security is becoming a hot topic around the world, this tool provides students the ability to learn from over 30 software applications that aid in penetration testing and more</td>
<td>Debian-based</td>
</tr>
<tr>
<td>Fedora</td>
<td>This OS is supported by the Fedora Project and sponsored by Red Hat. This OS provides a great resource for learning Red Hat Enterprise Language (RHEL). As there are thousands of jobs requiring expertise specifically with Red Hat, this OS is a great tool to prepare students for employment in IT. Fedora has over six Fedora Spins such as Design-suite, Scientific-KDE, Robotics, Electronic-lab, Games, and more</td>
<td>RPM-based</td>
</tr>
<tr>
<td>CentOS</td>
<td>This OS is derived entirely from RHEL. The source code is developed from Red Hat, which allows a student to learn RHEL with a small number of differences. CentOS can be used for teaching IT students on how to set up, administer, and secure a server</td>
<td>RPM-based</td>
</tr>
<tr>
<td>Ubuntu Studio</td>
<td>This OS is derived from Ubuntu. This OS is developed specifically for multimedia production such as audio, video, and graphics. Departments for multimedia could use this OS for multimedia instruction and the development of projects. As many of the tools for multimedia production are expensive, this alleviates large license costs for institutions</td>
<td>Debian-based</td>
</tr>
</tbody>
</table>
Communication and IT has also developed rapidly within this decade. With the enhanced role of IT, project managers must also prepare themselves to face the challenges of the future, both in the industry of project management as a whole and in the IT sector in particular. The increasingly important role of IT is undoubtedly the effect of rapid globalization that requires companies to have a larger capacity, timelier and more accurate information management within their decision-making system.

In project management part of a project manager’s toolkit is the Microsoft Office Project software suite. One of the many available OSS is Project Planner which allows the creation of Gant Charts, ability to manage resources, and ability to track tasks created. Using Linux-based software allows institutions and organizations to teach this critical skillset at a fraction of the cost (Fig. 3).
Additional OSS tools that are present can be found on websites such as Open Source as Alternative (OSALT), which provides open source alternatives for items such as Microsoft Windows Suite, Araxis Merge, Beaweblogic Server, and more (OSALT, n.d.).

DESIGN COLLABORATION WITH LINUX

In many graduate and undergraduate programs, institutions teach techniques for collaboration and system processes such as the Capability Maturity Model Index (CMMI). In teaching these techniques, OSS is the perfect platform as one of the key pillars for development is essentially coloration. With websites such as SourceForge, there are over 3.4 million developers participating in the development of over 324,000 projects. With these projects there is a group of developers that develop, deploy, and maintain these projects. For institutions this provides a method for students to understand the Software Development Life Cycle (SDLC) first hand.

Teaching and Understanding SDLC with the Aid of OSS Tools

The success of OSS demonstrates the alternative form of software and systems development processes. Software development is undergoing a major change from being a fully closed software development process toward a more community driven OSS development process (Deshpande & Richle, 2008). As a significant number of the information system (IS), computer science, or systems engineering students worldwide need to understand the SDLC, the OSS environment provides a great opportunity to learn all facets of the lifecycle (ACM, 2008). The SDLC known as the Waterfall Method (Fig. 4) is composed of six key phases that entail completing one phase and moving to the next without going back to the previous phase in the development cycle. The first phase is the requirements phase. During this phase of developments requirements are gathered for development. This allows developers and customers to capture requirements such as software functionality to user authentication requirements. The second phase of the SDLC is the design phase. In this phase the requirements become design specifications and developers develop the system. In the third phase, which is implementation, systems and subsystems are integrated to
function as an entire system. For example, a software application may be composed of multiple subsystems developed independently. At this phase they are integrated together for a final implementation of the system. In the four-phase verification, a final test of the software or system is conducted before the final release. In the fifth-state deployment the software application is installed after it has passed a rigorous check in the fourth stage. The last state is maintenance which consists of maintaining the software or system. It is important to also note that in this stage the decision to retire a software application or system is also made.

THE OPEN SOURCE LEARNING MANAGEMENT SYSTEM

Many studies suggest the traditional way of teaching, through books and static figures, appears to be inadequate in bringing across the complicated ideas of scientific concepts (Guimaraes & Murray, 2006). The current emerging media technology revolution supplemented the traditional face-to-face learning process with various e-learning communities as one of the fastest moving trends in today’s education to assist in preparing the students for more in-depth in interactive instructional environments could lead to enhance their learning opportunities in both online and on ground mixed-learning courses (Cole & Foster, 2007; Ko & Rossen, 2004; Rice, 2007; Waterhouse, 2005). Institutions around the world used various commercial teaching and learning applications such as Blackboard (http://www.blackboard.com) and WebCT to provide 24/7 communications.
between instructors and their learners. Those commercial applications could be very costly to buy their licensing.

In 2005, Munoz and Van Duzer mentioned that Humboldt State University paid approximately $8,600 for Blackboard license and they also pointed out that California State University system paid approximately 1 million dollars for licensing all the universities in the system. That could be huge amount of money for many institutes to licensing such educational systems. The best solution for such cost problem may lie in OSS (Wheeler, 2007). There are many open source programs available which are designed to solve numerous problems, but the most robust open source program is called Moodle.

Moodle is one of the most famous emerging media tools and widely known learning management systems (LMSs) in the U.S. universities, and other educational organizations (Martin-Blas & Serrano-Fernandez, 2009) used to develop a professional and more interactive educational (e-learning) environment especially for the higher education. Nowadays, Moodle comes as a first answer to the academic voracious demands for a professional and inexpensive tool for creating professional educational environment particularly by higher education and further education.

**Moodle Overview**

Moodle is one of the most user-friendly and flexible open source programs for electronic or e-learning (CoSN, 2010), identified as LMS, online interactive environment (OIE), and virtual learning environment (VLE) (Martin-Blas & Serrano-Fernandez, 2009), which has a large social framework of education support, and competitive alternative to many commercial applications. The word Moodle stands for “Modular Object-Oriented Dynamic Learning Environment, which is mostly useful to computer programmers”.

Moodle was developed from the ground zero by Martin Dougiamas who has great experience in both education and computer science (Cole & Foster, 2007) as an innovative e-learning tool designed to assist educators to easy build and share their courses online. Moodle was developed as an alternative approach to costly systems available in the market. It has variety of features and a relatively quick learning curve make it very popular tool among U.S. universities for creating online dynamic web-based teaching, and learning environment could be used as a stand-alone online
teaching and learning environment or as a supplement tool to their face-to-face traditional courses.

Dougiamas made the decision to make Moodle a copyrighted open source model to allow users to use, modify, add features, and distribute software package without modifying or removing the original license and copyrights (Melton, 2008). Originally, Moodle was developed for Linux OS but currently it is compatible with various OSs such as Windows and Mac. The first version of Moodle (Ver 1.0) was released on August 20, 2002. After the first release, there were many programmers all over the world starting exploring and examining the Moodle code, adding and removing features, and fixing the possible bugs if available.

Currently, there are many universities all over the world that adopted Moodle to build custom educational environments for their courses. Table 2 shows the list of top 10 countries prepared from registered sites in 223 countries.

Within academic environments, instructors can build their course specifying their course settings including the course format, course title, starting date, finish date, etc. Instructors can use Moodle to create stand-alone online courses by managing web-based content for their courses including course segments, lessons, focused technology, and so on. Additionally, they can use it to complement their traditional courses to facilitate complex courses’ concepts with limited face-to-face interaction, or they can use it to augment their traditional courses.

Moodle has many features including:

- easily managed courses;
- real-time collaboration and communication environment;

<table>
<thead>
<tr>
<th>Country</th>
<th>Registrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>12,710</td>
</tr>
<tr>
<td>Spain</td>
<td>6,558</td>
</tr>
<tr>
<td>Brazil</td>
<td>5,384</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>4,199</td>
</tr>
<tr>
<td>Germany</td>
<td>3,011</td>
</tr>
<tr>
<td>Mexico</td>
<td>3,007</td>
</tr>
<tr>
<td>Portugal</td>
<td>2,259</td>
</tr>
<tr>
<td>Colombia</td>
<td>2,154</td>
</tr>
<tr>
<td>Australia</td>
<td>1,808</td>
</tr>
<tr>
<td>Italy</td>
<td>1,758</td>
</tr>
</tbody>
</table>
• simple integrated quizzes and easy grade books;
• unlimited class size;
• unlimited enrollments;
• unlimited number of courses per school; and
• unlimited number of courses per teacher in particular.

In addition, Moodle provides many benefits such as

• 24/7 access from anywhere in the world to its learning environment;
• Upload and download course material including audio, video, .doc, .docx, PDF, image, and so on;
• Link to resources anywhere on the Internet;
• Easily create rich courses without need to learn HTML knowledge;
• Access files/papers/resources by a computer;
• Provide the ownership to the course content;
• Manage course content from year to year and never lose any work;
• Handle secure payments through using PayPal.

System Requirements and Obtaining Moodle

In particular, Moodle needs the most recent version of PHP, web application (one programming language), and SQL database such as MySQL or PostgreSQL. These tools could be found in one open source OS called LAMP. LAMP stands for the Linux OS, the Apache web server, MySQL, and PHP. In addition, Moodle should run an automated Cron process every 5 minutes or so (Melton, 2008). The host of the Moodle should include one. It is good to note that the Moodle website (http://docs.moodle.org/en/Install) has detailed information and complete instruction about how to install Moodle for the first time.

There are two options to obtain Moodle; the first option is that Moodle need to be downloaded and installed on a local installation on a personal server as part of a local network at the institution. This option could require highest maintenance. The second option is a remote hosting by a commercial web server. A user can contact one of the Moodle companies listed as a hosting at the Moodle services website (http://moodle.com). This process includes buying a domain name, and install Moodle software application on one of their servers (Melton, 2008). Those companies offered easy used tools for installing, upgrading, and maintenance. In this way, users will not have a highest amount of maintenance instead they will
have a mid of low-level amount of maintenance, but those companies will charge a fee for doing that.

**Moodle Organization**

Moodle could be set up according to its needs to deliver the information to the audience. For example, it could be used either as a networking environment or for handling course purposes. Instructors need to choose the format of the course by having only one screen that either contains the course material or consists of semester, term, or even year. In addition, the right and left sides of the main central course content could be arranged to include upcoming events, open forums, course members, latest news, search, recent and upcoming activities, etc. ([Fig. 5](#)). Moodle organization’s layout could be edited by administrators who have skills with Hyper Text Markup Language or HTML code and cascading style sheet or CSS. In 2008, Melton explained Moodle setup items as follows.

![Screenshot of Moodle Course Page.](image-url)
Moodle organization’s layout can be edited by administrators who have skills with HTML code and CSS. In 2008, Melton explained Moodle setup items as follows:

1. Forum: It is asynchronous module. Instructor can set up four types of forums such as standard forum, a question and answer forum, a forum for each member, and a forum for just one discussion. Instructor can use these entire forums to generate discussions, set for grading, assigning deadlines, attachment files, block attachment, email, and more.

2. Group: Instructor can make team tasks by combining a certain number of students to do a certain project(s) or any other course work. Instructor can hide the group work from other group and make it visible for the other group members by the end of the course. This allows students to learn how to work in virtual groups which according to a survey by Wainhouse, 72 percent of the respondents are more interested in alternatives to travel since the September 11, 2011 event (Wainhouse, 2002).

3. Grading: The latest version of Moodle allowed educators to assign grades for students and export the grades in different formats. These tool can be utilized to allow the student and the instructor to track progress throughout the course.

4. Assignment: Instructor can use this area to organize his course assignments in different ways and assign a specific deadline for submitting each assignment. Moodle allows students to submit their assignment in different formats such as text file, PDF file, Word document, and PowerPoint presentation, Excel sheet, pictures with various formats, etc.

5. Quiz: Instructor can set up their quizzes for each student individually or in groups. Instructor can open and close each quiz in specific day and time, send a customize feedback for each student, and so on. Quizzes can also be developed from a random test bank to minimize cheating.

6. Chat: It is a synchronous module for communication allowing the instructor to announce chat times, groups, and session logs. Student can use chat for real-time communication with or without their instructor for specific discussion issues. These chats can also be archived to be viewed at a later time to review the communication to ensure the message was understood or relayed appropriately.

7. Message: It provides both synchronous and asynchronous benefits. For example, a message window will pop up when the student change his web page during sending the message process. On the other hand,
8. Blog: A communication module located on a user’s profile. Users can use their blogs for personal publications, read and write current entries’ purpose, and access popular topics easily. The publications appear in reverse order, so the most recent post appears first.

9. Wiki: A communication module could be used in a different way. Normally users use Wiki to collaborate with each other on coursework and projects. They can use it as a brainstorming session to exchange their thoughts about ascertaining subject. Wikis can also be exported to reside on a student’s or professor’s desktop. Wikis can serve a living document which can be updated at any time to include serving as configuration management systems for manuals or books.

10. Lesson: Instructor can use this module to prepare lessons in multimedia or PDF file forms. Also, the instructor can add questions to the content to check his or her understanding to the subject.

11. Glossary: This is very important module that can help the instructor to explain the difficult vocabulary word to the students and add new concept to the content to facilitate the concept of the course content.

12. Gathering feedback and data: Instructors can use this module to gather student’s feedback and/or other types of data in three different ways: database, survey, and choice. This module can be exported for a review or to provide a synopsis of issues to administration.

13. Administration issues: Site administration panel contains many tools that allow the instructor to create and organize his course(s) with respect to the privacy and security issues. For example, they can create new courses and share them with the others and store them if necessary. In addition, they can make a manual or automatic backup for their courses including course data and student data (Melton, 2008).

Moodle has associated weakness. Moodle does not have the ability to reorganize courses into functional groups for each user. It only has a few related organizations by the topic of the current and previous courses and weekly course descriptions. The user would be blind and does not have the ability to see his courses’ organization and could miss his opportunity to reorganize his courses.

Moodle does not have any link for advanced conferencing features such as a whiteboard conferencing feature. Sometimes chatting with the professors alone does not have the ability to convey the exact meanings like...
visual information always do. So this facility allows the professor to visually share information in real time and allow the other students to watch the professor’s movement and discuss with him.

Summary

The Moodle is great online learning software or course management system used by a wide variety of users and allows professors and students to conduct and participate in an electronic classroom. Students have the ability to post their discussion items, submit their assignments, and post journals and resources as attachments.

Moodle has a simple interface, uses a minimum of words, and includes simple icons with the words to aid users and show them the direction to where he can find the information they looking for.

REVIEW OF VIRTUAL WORLDS IN THE CLASSROOM

Emerging technologies offer exciting new teaching options for teachers and new learning options for students (Al Saeed, 2011a). For example, open source as one of the emerging media popular tools offered free innovative base to develop new teaching tools for instructors and new learning environment for students. In 2011, Dawson argued, “Simulation-based training provides many benefits to education, commercial, and military industries, allowing trainees to test and hone their skills in a safe virtual environment” (Dawson, 2011).

Financing could be the main driving forces behind utilizing open-source software development especially for the institutions of higher education (Miro International Pty Ltd, 2006; Wang, 2004; Rooij, 2007). Open-source virtual simulator software applications such as Second Life (SL) and Open Simulator (OpenSim) as one of the most popular virtual world environments might be considered as a potential solution to eliminate financial restrictions could face any higher education institution (Pan & Bonk, 2007). Al Saeed argues that “SL and Open Simulator (OpenSim) provide an opportunity to create innovation space educational simulations and allow students to become immersed in the learning process and help in convert them from passive to more active learners to become hand-on operators in more interactive environment. In other words, simulation
environment could be designed using emerging virtual technology for educational purpose such as space education purposes. It may turn students to be active learners by engaging them in an immersive virtual environment” (Al Saeed, 2011a). University can easily access to source code, modify it, save the license fees could be chard by the vendor to develop and provide the institution with the flexible and cost efficient learning environments (Pavlicek, 2000; Weber, 2004; Williams, 2002).

Virtual world such as SL is a public utility for digital media that includes many features such as 3D graphics and simulation technology, text chat, voice chat (VOIP), and advance digital media for online collaborative courses and classrooms’ environment.

Based on the emerging global distribution for web and computerized technologies, all types of virtual worlds were designed on the existing web standards to create a global virtual social network for socializing, messaging, media delivery, storage, and processing. It is an open and extensible platform that enables of Immersive education and distance for a wide range of variety students.

SL as one of the most popular virtual world provides a unique and secure teaching and learning environment for instructors and students. This learning environment could be designed to be private for a specific school where each avatar assigned to a unique name could include the name of the university, collage, or any other educational institution that the individual belongs to.

Many advances are incorporated into the design of such educational environment using virtual world such as SL. For example, the universities that host their own servers have complete control over who

1. Has access to their educational virtual worlds and has the ability to restrict access to those worlds at their discretion. For example, there were no anonymous access allowed for virtual worlds hosted on the Education Grid. Universities can also provide student rosters, and teacher rosters, from which authenticated avatars may be provisioned.
2. Has the ability to keep their virtual worlds for education private, or make them available to some other learners.
3. Has the ability to make separate copies of a virtual world simultaneously available for few levels of learners, where appropriate security measures are applied to enable students and teachers to work together in safe and secure learning environments.
4. Has to ability to have a single virtual world be utilized, simultaneously by multiple universities. Additionally each school can be provided with
its own private audio channel, which prevents lectures and discussions from being heard by anyone outside of that particular class.

Higher education technologists especially for those in the doctoral/research institutions utilized the efficiencies of open source. Colorado technical university is one of the first universities who realized the importance of using virtual worlds in teaching. They designed their own virtual campus as new teaching and learning environment in SL (Fig. 6). Additionally this institution has developed new degree programs at the graduate levels to teach emerging media and technology (Al Saeed, 2011b).

**SL OSS Educational Competitor**

One of the virtual world education grid competitors is OpenSim. OpenSim is an open source multi-user 3D application server designed by taking the advantage and make a reverse-engineering to the published application programming interface (API) functions and specific Linden Lab open source parts of the SL code. One of the strength for creating any virtual environment is making it accessible by a variety of users through using various protocols. OpenSim offers a great opportunity for the virtual world developers to create customized virtual worlds easily extensible through using the technologies that fit with their needs.

One of the weaknesses is that this kind of virtual worlds are susceptible to attacks by regular web clients. It is actually easier to copy assets with a
web-based client. The weakness is that asset servers are connected to the public Internet, and the protocol for interacting with them is public.

The Virtual World Education Grid will result in an improvement over the OpenSim because it has a great potential in language learning and teaching, projects turn out to be motivating both for the students and instructors, task-based activities are favored over other methodologies. On the other hand, OpenSim is high technological requirements and hard user interface compared to the virtual world grid.

**Virtual World Educational Grid**

Demand for online or non-traditional classes is increasing day after day. Technology is providing more tools in which schools can enhance performance of their learning model. The Virtual World Educational Grid is considered a free environment developed upon existing web standards in order to emerge a unique environment for virtual world delivery.

Virtual World Educational Grid has many powerful features including interactive and collaborative 3D graphics, game technology, audio, video, simulation technology, VoIP, text chatting, web cameras, and so forth with collaborative online course environments. The educators and students will fell themselves are physically attending a class session even that is not possible and that will cut any cost associated with attending class sessions in real world. Students from all over the world can communicate with each other in a way they can enhance their learning experience process. It will engage the instructors, educators, and students with the teaching and learning process in the same way they can collaborate within the traditional teaching and learning process. Virtual World Educational Grid supports individual-based learning process as well as group-based learning environments where the interactive lessons could inject into larger bodies of course material in order to enhance the education learning and teaching experience.

**Identified Baseline Requirements**

1. Accessible: The Virtual World Educational Grid should be designed in a way to be accessible to all users and institutes thought to previously set up a specific procedure and open usage policy, but also preserve the right to make the server private for events.
2. Open Application Programmer Interface functions and a portion of the code: The Virtual World Educational Grid should be flexible and extensible for virtual world developer to make customized virtual environments, and that would be possible by publishing open API functions, protocols, and a portion of Virtual World Education Grid code.

3. Open and interoperable file formats: Software configuration and files should be transferable to other environments and platforms with respect to the minimum hardware needed.

4. Open hosting with conformance and compatibility: The Virtual World Educational Grid should be designed with the relevant educational and operational standards to achieve the conformance and compatibility standards.

5. Support multiple content formats: The Virtual World Educational Grid should support 3D content in addition to other content formats such as gaming, audio, video, text, etc.

6. Quality control: The Virtual World Educational Grid content should be reviewed and categorized by using metadata in order to meet the standard and be accepted into the Virtual World Educational Grid. That includes qualitative analysis, rating, and tagging by educators, students, and expert people of the subject matter.

7. Security and privacy of the learning environments: Developing Virtual World Educational Grid requires variety levels of protection. Learning environments could be created with some levels of security and access procedures for those only who are authorized in order to provide safe and private learning environment for the learners.

**UTILIZATION OF VIRTUAL WORLDS IN ACADEMIA**

Currently, virtual worlds are widely used for training and education purposes to facilitate trainees’ learning activities (de Freitas, 2008; Dobson et al. 2001; Granlund 2001; Robert et al. 1996). Virtual worlds allow users to create their own avatars, which are referred to their residents and offer them the platform they need to interact with each other easily. Virtual world is a playground for imagination and expands the boundaries of users’ creativity in exploring, defining, creating, designing, modeling real environments, building, coding, document sharing and recording facilities, performing, and collaboration (Chen et al., 2008). Virtual worlds can be very effective and cost-efficient
environment that can provide a new methodological framework that supports training purposes to include serving as a tool for collaboration. Virtual worlds are a playground to simulate real-world applications for training purposes, but there is a chance it may not be able to simulate overall scenarios with small details that could be involved in real system because of the limitation boundaries of virtual worlds (Stolk et al., 2001). This chapter lays out the strategic use of virtual worlds for training purpose and provides a framework based on existing methodologies could be employed to facilitate modeling and testing process.

SL is an associated free client program known as the viewer which allows the users, known as residents, to interact with other individuals through avatars. Students created avatars which were representative of their personality and how they best felt to interact with other residents. This environment can be described as environments created by technology that have incorporated virtual representations of elements found in the real world (Kock, 2008). This course allowed students to develop virtual objects with limited knowledge of the Linden Scripting Language. As the Linden Scripting Language has syntax familiar with the C programming language, the students were able to jump right into developing scripts. Fig. 7 is an example of the Linden Scripting Language that would be utilized to have an object display text when they are five meters near the object. In terms of SL the most common method to gather information is through blogs and special journals as this knowledge base is constantly growing (Kern, 2009).

OpenSim is an OSS application that can be utilized to simulate virtual environments. This would allow architects and designers the freedom to develop their own virtual representation of an environment. However, an architect would need to have knowledge of the .net framework and programming in order to develop or modify items without any support. Furthermore, this competition not only developed a health care facility, but also created an entirely new vernacular by approaching this with SL virtual design capabilities. The introduction of a new form of instant communication beyond words and diagrams but in a full on demonstration of ideas creates a platform for creative designs. The implication of SL into competitions will provide user with a clear path to think creatively. The ideas can then overflow into a multi-faceted direct source, instead of just using one or two types of devices to demonstrate ones ideas, such as through email, telephones, video, or webcasting. The impact the SL has had will grow into a more influential design world.
University Graduate Projects Using Virtual Worlds

The introduction to Advanced Systems Engineering course at Morgan State University's School of Engineering graduate students participated in a group project that consisted of creating a virtual education center using SL, a virtual world developed by Linden Lab (Dawson, Burrell, & Emanuel, 2011). This project permitted students to apply their systems engineering concepts and skills learned as a result of the provided course materials (Dawson, Burrell, & Emanuel, 2011). The initiation of this course project was through a Request for Proposal (RFP) provided by the course professor who acted as the end customer. Displayed in Fig. 8 is a screenshot taken of the IEGR 501 SL Class during a live session in a virtual world during the Fall 2010 semester.

The goal of the virtual education center was to foster learning allowing for the development of engineering capabilities, and to hold large...
conferences of approximately 100 researchers. The land consisted of multiple regions. The first portion of the land was required to provide a large conference center that allowed authentication to occur for all entering personnel. The second portion of the island was to act as an engineering development ground. This development ground was to consist of objects available for testing. Also, in the engineering, development ground was a rapid prototyping center for developing many applications related to the present world. The third portion was to be a classroom environment for
three classes to be held simultaneously without hearing the other classroom during the lecture.

Throughout the semester, every student worked diligently to adhere and produce to the list of requirements proposed by the professor. Small groups were developed within the team to formulate a list of items needed and their pricing. The necessary items ranged from land to office equipment. After all of this preliminary research has taken place, the professor chose the most feasible items presented from each group, purchased them, and put together this virtual education center using SL.

**Systems Modeling Using Virtual Worlds**

Beginning from the initial problem definition and users’ requirements, a High Level Systems Analysis (HLSA) was proposed resulting into graphically modeling the system with high-level and low-level system diagrams. This allowed the developers to capture the main important entities within this project.

Once analyzing the problem and system, a plan for design was implemented into Enterprise Architect (EA) using the SDLC OOAD methodology. The idea then leads to purchasing objects from the SL market to use as prototypes within the environment of SL. Difficulties aroused while using the software SL, such as programming objects as well as receiving objects from the market; some objects were unable to be modified and required to purchase other objects that would cooperate with the proposed system and environment.

The overall experience was interesting in learning to plan a development of a graphical user interface (GUI). Future work will be to present the development process of the research project as well as further enhance knowledge within SL to use an effective tool in simulation work.

**Optimizing Student Participation in the Design Process**

A widely held principle in the field of systems engineering is that the success of a system is directly proportional to the extent of user participation in developing the system. The results of this are that when an end user has a higher level of perceived meaningfulness task this would positively impact subjects’ attitude and performance. The other result is that when the user has a perception of control and procedural justice then the user’s outcome is satisfaction and their performance increases for the subjects as one increase the
user’s mode of participation. Perception of user control with procedural justice yields an outcome that increases the performance of a project as the user is given more opportunity to voice their opinion. It is interesting that when the users are given the choice to establish boundaries in the decision-making process there are increasing gains in user participation attitudes and performance. When the user’s meaningfulness task is increased then procedural justice and control task commitment and performance also increase. The user participation positively influence perceived control even though none of the users received their preference. Perceived control influence perceptions of procedural justice. The direct path from decision control to outcome satisfaction reinforces the fundamental importance of perceived control. Path analysis demonstrates a direct effect of task meaningfulness on performance. Meaningful task evokes feelings of inclusion and increase perceptions of control.

Modeling and Designing Using Virtual Worlds

In this section, two design projects shall be described in detail. The first design project shall be a military training facility and the second shall be a modern student building. Both projects were designed as part of graduate level coursework in computer science and systems engineering programs where the students had limited knowledge on architecture and design principles. The students were allowed to use script generators to develop code for objects to include purchasing some of the materials needed from the SL marketplace.

The utilization of SL for these graduate students provided them the ability to develop concepts with the larger picture in mind. As many new computer science and system engineer graduate students are new to the concept of systems thinking, these projects provided them the ability to understand how design concepts are essential to the larger picture. Many of the students thought that job was only important for architecture, civil, or design students; however, they quickly realized the importance of understanding these design concepts themselves in order to provide a full solution to the end users or customers.

Virtual Military Training Facility

This design project was to develop a military training facility that would be the virtual representation of a training facility that could be utilized in
the real world (Dawson, 2011). This virtual prototype was developed to test the usability of a proposed military training facility. The facility required a passcode to enter to include a presentation viewer that displayed training on national security. This design project went through the systems development process. Requirements were captured and tested during the design of this system to include a significant number of tests conducted with live participants. Fig. 9 displays the end result of the completed military training facility.

This design project proved that SL was a viable place for designing military training facilities. SL also proved to be a cost-efficient tool for testing the system’s design concepts and usability with end users to include stakeholders.

Modern Student Building Project

The main objective of the modern student building project was to research and clearly understand the requirements of an automated teller machine (ATM) system, as well as understand the application model requirements for integration with an ATM system simulation. The goal of this research
design project was to discuss the planning, analyzing, designing, implementing, testing, and evaluating phases of the development of a GUI of an ATM machine model using the software SL; virtual world (Davis & Dawson, 2012).

To accomplish this design project the researchers had to plan the path in which a developer will take to follow for production. This meant properly analyzing requirements and literature review to understand the entities within the system. The use of Object Oriented Analysis & Design (OOAD) to graphically model users, use cases and scenarios, data and flow diagrams was implemented.

The Object Oriented (OO) model was implemented into SL by constructing an environment in which the system will possibly be able to operate in (Davis & Dawson, 2012). Programmed objects to function when virtual users wants to perform a task. Fig. 10 displays the system modeling done for the project.

Beginning from the initial problem definition and users’ requirements, a High Level Systems Analysis (HLSA) was proposed resulting into graphically modeling the system with high level and low level systems diagrams. This allowed the developers to capture the main important entities within this project.

Once the program was analyzed and system a plan for design was implemented into Enterprise Architect (EA) using the SDLC OOAD methodology. The idea then leads to purchasing objects from the SL Market to use as prototypes within the environment of SL. Difficulties aroused while

**Fig. 10.** Low Level Systems Diagram.
using the software SL. Difficulties such as programming objects as well as receiving objects from the market; some objects were unable to be modified and required to purchase other objects that would cooperate with the proposed system and environment.

The overall experience was interesting in learning to plan a development of a GUI. Future work will be to present the development process of the research project as well as further enhance knowledge within SL to use an effective tool in simulation work. Fig. 11 displays the finished design project in which live participants utilized the system to capture valuable data to test the hypothesis.

**FINAL REMARKS**

This chapter discussed the utilization of Linux, OSS, and virtualization tools that are currently available. Also discussed was how OSS virtual tools have been applied at the university level in graduate course projects to teach SDLC. We have described the benefits and challenges of OSS implementation to include how some countries are currently using OSS.
It is important that we as educators utilize these tools to allow students everywhere the opportunity for creativity and learning. As we realize that every institution worldwide does not receive the same level of funding, it is imperative that these tools become part of a technological arsenal to raise educational experiences with minimal impacts to funding.

ACKNOWLEDGMENTS

We would like to recognize Dr. Cynthia Calongne who is a Professor and Chair of Emerging Media in the Doctor of Computer Science and Doctor of Management degree programs at Colorado Technical University who provided great mentorship in the development of virtual worlds. We would also like to recognize the system engineering graduate students at Morgan State University and faculty member Dr. LeeRoy Bronner. Lastly, we would like to thank our families for having patience during the completion of this submission.

REFERENCES


