MIST/C: OPEN SOURCE SOFTWARE FOR HYBRID CLASSROOM AND ONLINE TEACHING

J. Mark Pullen, Nicholas K. Clark, and Priscilla M. McAndrews
Networking and Simulation Laboratory
George Mason University
Fairfax, Virginia, USA
{mpullen, nclark1, pmcandre}@netlab.gmu.edu

ABSTRACT
Educational research indicates that the two prevalent approaches to distance education, synchronous and asynchronous, exhibit synergy if combined. Our research addresses how to create the most effective, open source (and therefore robust and inexpensive) software to enable educators to exploit that synergy. The Moodle learning management system provides a high-quality asynchronous online learning environment, but until recently has lacked a matching synchronous component. This paper introduces synchronous open source software for filling that gap: a synchronous online teaching/conferencing system called MIST/C. The Moodle-MIST/C combination has demonstrated its effectiveness in online delivery of our Master of Science in Computer Science Department. MIST/C was created as an enhancement to the Network EducationWare system, on which we have reported in previous IASTED conferences. It was developed as open source software to work seamlessly with Moodle on all major platforms and also to be easy to install and operate. It also is tailored to support *simulteaching*: teaching a group of students in the hybrid mode delivering to classroom and online at the same time, which conserves teaching resources. This paper describes the design and implementation of MIST/C, including latest advanced features, and our experience using it to deliver our degree program.

KEY WORDS
Internet distance education, hybrid synchronous/asynchronous

1. Introduction

Distance Education (DE) approaches generally are categorized as either synchronous DE, where student and instructor communicate electronically in real time, or asynchronous DE, where the instructor commits the learning materials to some medium, from which the student receives them afterward. It is recognized that these two approaches are synergistic; they are complementary and can enable better effect together than separately [1][2]. However, the open source educational software community has been slow to produce systems combining the capabilities. Our research addresses how to create the most effective, open source (and therefore robust and inexpensive) software to enable educators to exploit that synergy.

The authors have been engaged for the past decade in implementing open source DE software for synchronous online education, as reported in [3,4,5,6]. Moreover, we share the observation of [1] that synchronous and asynchronous modes are more effective when combined. When we discovered the software integration facilities in the Modular Object-Oriented Dynamic Learning Environment (Moodle), we were motivated to rework our previous efforts into the Moodle environment. Moodle is widely recognized as a high-quality Learning Management System (LMS) supporting asynchronous DE [7], with over 49,000 currently active sites registered in 211 countries [8]. Our previous product, Network EducationWare (NEW), integrates a collection of open-source synchronous communication tools into an online teaching environment [9] that also has been used to a significant extent for Internet conferencing [10]. Because of its low cost and user-friendly interface, we already were using Moodle as an LMS for course delivery supported by NEW; upon discovering the developer-friendly facilities offered by Moodle [11], we factored Moodle into our development plans. This paper describes the result: Moodle Integrated Synchronous Teaching/Conferencing (MIST/C) [12].

2. Background

The capabilities of NEW are described in [5]: audio, video, and whiteboard interfaces, a floor control/chat panel, a recorder, and a playback unit, all operating under control of a master client. The system can function well (absent video) over a 56 kilobit/second modem, and even better (including video) over a good Internet connection. From 2002 to 2009 we continued to enhance the functionality, performance, and reliability of the NEW open source software. This included extending NEW from the Windows platform to Linux and MacOSX and building the comprehensive NEW Course Management (NEWCM) system for file maintenance and user authentication that won an award for non-commercial, Web-based educational software [13]. It also provided an ever-growing suite of documentation that supported use of NEW by fifteen faculty members teaching about twenty different courses, mostly at the graduate level, a program well suited to our region's need for telecommuting to class. Particularly notable was the infrastructure, both computer and human support, that provided for libraries of teaching slides and lecture recordings, captured as a by-product of using NEW to teach in a normal classroom setting. Use of NEW to support international teleconferencing in a software development program demonstrated its versatility [10].

3. Synchronous Teaching Rooms in Moodle

Moodle is open source software, designed with a modular architecture. It supports a plugin application programming interface that makes enhancing Moodle's capabilities or adding features relatively straightforward. There is a library with hundreds of plugins available from the Moodle community [14]. MIST/C was developed to use this plugin interface as a Moodle "activity module." As a result, it is very simple to install (or uninstall) the plugin in an existing Moodle environment. Installing the plugin automatically creates the required database tables and entries to support the MIST/C server software. By integrating with Moodle, MIST/C is able to take advantage of existing Moodle infrastructure, thereby avoiding the need to maintain a Web application for file management and user authentication. This allowed us to focus our development effort on improving the synchronous software components without maintaining a separate Web interface.

Most educational institutions maintain systems for managing student accounts and enrollment using technologies such as the Lightweight Directory Access Protocol (LDAP). Moodle is built to take advantage of such systems; MIST/C inherits this capability. The Moodle *student*, *instructor*, and *administrator* roles provide appropriate privileges for synchronous class sessions. Instructors and administrators have the ability to control the "floor" in the context of the online class, determining who can send audio or manipulate slide content. Moodle also has extensive capabilities for managing content and organizing it in formats appropriate to the type of course. Courses can be organized into weekly, topic-based, social, or SCORM-based formats [16]. Using Moodle's file management capabilities, instructors can arrange their MIST/C slides and recordings in the way best suited to their course format.

MIST/C is a client-server software system. The MIST/C plugin adds a new activity module called a "MIST/C Virtual Room" to Moodle [17, 18]. Instructors can have a synchronous capability for their course simply by adding this activity, using the Moodle editing interface. In the context of the synchronous class, a virtual room is a dedicated meeting space used only for a specific course, which is available for use at any day and time. In the context of the software architecture, a virtual room also is a specific server process, which must be running on some networked computer that is reachable by the MIST/C client.

The MIST/C Moodle plugin is available on our website as a virtual machine module, complete with Moodle and all supporting open source components, making server installation very straightforward in a VMWare environment on all major platforms (MacOSX, Linux and Windows). Clients for these platforms also are available on our website [12]. The source code for MIST/C client and server also is available there.

4. MIST/C Advanced Features

We began the development of MIST/C by working with the Volgenau School of Engineering Distance Education Committee, to define all features sought by our teaching faculty in a hybrid synchronous/asynchronous online teaching environment. Tables 1 and 2 give the features considered important by that committee. (At the time of the review, only one commercial product met all of these criteria: the *Elluminate* system, now known as *Blackboard Collaborate*.) The MIST/C client and server were designed to fulfill these requirements, by incorporating a number of improvements over the predecessor system NEW. A description of the enhancements follows.

Table 1. MIST/C Hardware/Software Requirements

Client Operating System: Windows XP/7; MacOS X; Linux
Browser: all major
Communication: wired/wireless Internet and dial-up
Security: LDAP authentication
Limitations: seats and classes unlimited
Remote hosting: available at moderate cost
Availability: simple download
Audio: Internet; no separate phone connection needed
Added hardware required: mouse/keyboard/mic; WebCam
Responsiveness: rapid response to user inputs

Table 2. MISTC Functional Requirements

Customizations: accessible/expandable/enhanceable
Whiteboard: able to accept graphic files in real time
Authoring formats: PowerPoint, PDF, Keynote, OpenOffice
- all participants able to annotate slides during session
Video: common computer formats (mpg, avi, mov); camera
Recording of sessions: automatic on server; including chat
- able to render as mpeg for podcasting
Interaction: testing; polling/hand-raising; voice + chat
Student tracking: login status; participation statistics
Configurable to screen: by user; application window capture
Breakout: able to partition class into separate groups

4.1 Auto-reconnect

The MIST/C client can operate in an offline mode for viewing and creating recordings. However, for live classes it operates by connecting to a remote server on a network, usually the Internet. All network connections are not created equal; even with the better quality ones, transient failures can happen. Even with an excellent network environment, broken connections happen sometimes. Our courses using MIST/C or NEW operate in a mode that we call *simulteaching* where instructors teach both a classroom group and remote students at the same time. In this mode, a network disconnection that interrupts the class and requires the instructor or a support person to restart the software can be very disruptive.

To address this issue, MIST/C supports an auto-reconnect system. The Master Client detects the network disconnection, notifies the instructor, and attempts to reconnect to the server. The reconnection happens automatically and does not interfere with the face-to-face live class or the recording. In rare cases where the client cannot reconnect due to a serious network issue, the instructor can continue to teach, producing a local recording to be posted for online students to download from Moodle later.

4.2 Server-side recording and download

One of the most useful features of MIST/C is the ability to record all of the online activity in a live class session. This includes audio, chat, video, and whiteboard slides with real-time annotations, all fully synchronized. With NEW, recordings could be made by the instructors (or students) locally by pressing 'record' in a recording control interface on the client software. The control includes the capability to pause the recording, which is normally done during breaks. Instructors value this feature; however, being human, they sometimes forget to start the recording or fail to resume it after a break, resulting in missing or incomplete recordings. To avoid this problem, MIST/C includes a capability to automatically create recordings of class sessions on the server as well as the client. These recordings can be retrieved through the instructor's client and posted for students in Moodle if the client-side recording is inadequate.

4.3 Interface improvements

The NEW user interface included one independent window for each active component: audio, video, whiteboard, floor control, record control, play control, and master client. All of these windows were managed individually, creating confusion when they were minimized or misplaced on the desktop. In MIST/C, all of the controls are integrated into a control window with a reduced footprint on the screen and a set of toggle buttons to display or hide all functions except the essential audio. This makes using the component interfaces simpler and allows more space for slides, floor control, text chat, and video. It also supports a second whiteboard for the classroom projector so that students see full-size slides, uncluttered by the other components. The Floor Control has been upgraded to display each user's full name, where NEW displayed only an eight-character username. As components are started and stopped, the interface automatically resizes and also relocates the Whiteboard and Video windows. Figure 4 shows the complete MIST/C interface, without the video client. (We generally do not use video in teaching Computer Science.)

4.4 Added whiteboard functions

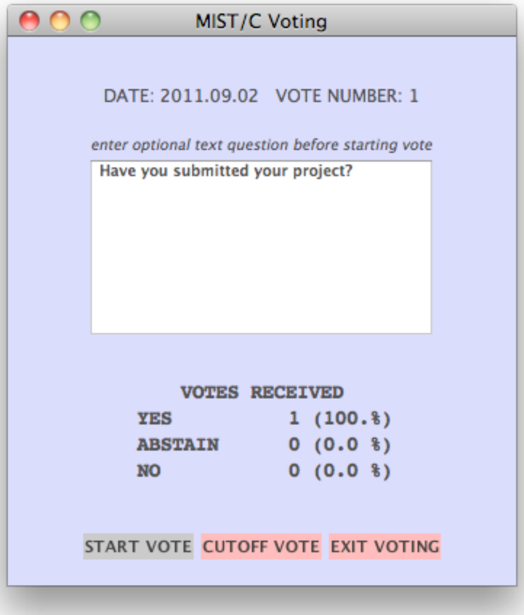
The two features of MIST/C that are essential to most teaching are the audio interface and whiteboard, both of which operate by streaming data over the Internet among participants. All users have the same capability, but only one user can send audio and whiteboard at any given time; which user can send is determined by the floor control function. Thus, subject to the instructor's control, students can speak to the group in the classroom and online and display their own slides, a feature we use for student project reports and discussion-based seminar classes.

The whiteboard displays a static slide plus dynamic, multicolor annotations made by the presenter. Originally the whiteboard supported only single-page PDF and PostScript formats. We added JPEG to this to allow NEW to send a compressed, if somewhat fuzzy, slide to online students quickly. Also added was the ability to convert OpenOffice and Microsoft PowerPoint slides to these formats automatically. Under MIST/C, we have added the ability to import multi-page PDF slides, providing a generic interface for all authoring software. The MIST/C whiteboard also has the ability to import PNG slides, which are crisper than JPEG, and to import to the whiteboard the window contents of any application running on the client computer. Window import is particularly useful when presenting a software application to the class. It features options for both one-time and automatically repeated import.

The whiteboard has a feature added specifically to support simulteaching: it is possible to configure a second, read-only copy on the teaching computer, for use with a classroom projector. Content of the second whiteboard mirrors the one used for input by the instructor, but the control buttons are eliminated. As a result, students get a full-screen view of the slides while identifiable student information is shown only to the instructor, which increases online student privacy. This capability is not present in any other synchronous teaching system of which we're aware. When we've used a commercial system to teach, the whiteboard has been only about one-third of screen area while the identity and comments of online students have been projected in the classroom.

4.5 Voting

A common classroom request from the instructor is "raise your hand if…"; MIST/C supports the online equivalent by a Voting mechanism. The instructor selects the Voting interface by a button on the Floor Control, which results in the window of Figure 1 showing on all participants' desktops. The question being voted on can be typed into a text window, or simply spoken. When the instructor selects "START VOTE," the question appears on student Voting Panels and students may enter votes. They also can change their vote until the instructor selects "CUTOFF VOTE," at which time the vote is frozen and results are written to the log.



DATE: 2011.09.02   VOTE NUMBER: 1

enter optional text question before starting vote

Have you submitted your project?

```
VOTES RECEIVED
YES          1 (100.%)
ABSTAIN      0 (0.0 %)
NO           0 (0.0 %)
```

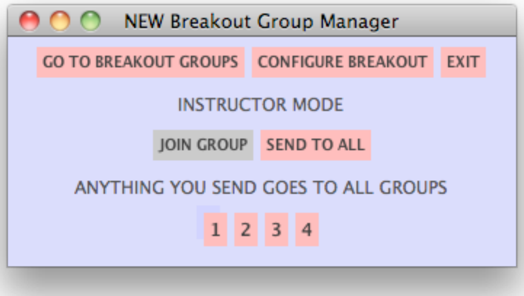
Figure 1. MISTC Voting Window

4.6 Breakout rooms

Some learning topics are dealt with better by breaking the students into discussion groups. MIST/C supports this situation by allowing the instructor to open the Breakout Group Manager (Figure 2) by a button on the Floor Control and assign students to groups. The instructor can then select "GO TO BREAKOUT GROUPS," whereupon all students can communicate within only their assigned group until breakout mode is cancelled. In this mode, the instructor can interact in one of two ways:

1. Join a group and communicate only with the members of that group

2. Stay in an oversight role; this allows one-way communication, where the instructor's voice and graphics are displayed to all students if the instructor takes the floor, but student replies are not possible



INSTRUCTOR MODE

ANYTHING YOU SEND GOES TO ALL GROUPS

Figure 2. MIST/C Breakout Manager

5. Teaching with MIST/C

In our experience, encouraging students to test your software is a good way to find problems. In the Spring 2010 semester, 27 students in a graduate Computer Science course tested MIST/C as part of classes taught in the simulteaching style. Moodle was used as the LMS and was combined with MIST/C for attending class in real time and reviewing recordings. Homework assignments and lecture slides were posted in Moodle. As a project in the class, 17 students chose to use MIST/C to create their own oral reports including audio, slides, and annotations. The reports were recorded by the students independently, using MIST/C in offline mode with no assistance. In addition to recording the lectures, students provided feedback on their experiences with the software. Of the 17 students who attempted the project, all successfully produced presentation recordings. Most of the feedback was positive, although several students reported on a specific documentation problem that confused them. We modified the software to incorporate their suggestions before expanding its use to the full online graduate teaching program.

In Fall 2010 and Spring 2011, MIST/C was used by all twelve online courses in graduate courses taught by the CS department, ranging from Introduction to Algorithms to Software Project Management. Teaching styles ranged from traditional lecture to project-based seminars. The experience generally was very positive. We did experience some problems with multi-page PDF conversion; it turns out that there are several ways to implement the Abode PDF specification and many authoring applications now have different implementations. However these problems were resolved and all faculty members had a good experience teaching with MIST/C in its first full semester.

6. MIST/C in a degree program

The GMU Department of Computer Science has offered online courses since 1995 and has offered a full Master of Science in Computer Science degree online since 2004 [20][21]. Teaching faculty tailor their use of delivery technology to their subject and individual style. As a result, there is significant variation among the 17 courses now offered online. However, it is possible to generalize that all courses are offered in some variant of simulteaching mode, and all have been offered using MIST/C.

What we believe to be a non-issue is the question “which is better, online or classroom delivery?” Research shows that, all other things being equal, there is no significant difference in student learning across a wide range of delivery methods and technologies [22]. This is borne out by our own data collected over five years, as shown in Figure 3.

We believe the appropriate question is “How can we best help the student to learn?” Clearly, the answer to this is highly dependent on the student’s learning style and personal situation. For example, two situations, common among our students, contribute to their enthusiasm for online course delivery:

- Our school is in a major urban area that is well known for traffic congestion. Our graduate courses are delivered in after-work “rush hour” time, with the result that commuting to and from class can require as much of a student’s time as does the class itself.
- Many of our graduate students are employed in government and industry and must travel as part of their work, but can either connect remotely or attend class “time-shifted” via recordings.

We find that offering students the choice between online and classroom delivery lets them adjust most effectively to their situation and learning style.

6. Future Work

Our latest work on MIST/C is aimed at the ability to scale operations from tens of courses to hundreds, in a cloud environment. Managing such a large collection of MIST/C servers effectively on a scheduled basis would be very difficult; a much better arrangement is *dynamic load balancing*: when a MIST/C session starts, a scheduling process will choose the available server platform with the lightest load and designate that platform to support the session. We have prototyped a MIST/C server in the Amazon Elastic Cloud [18] and determined that a straightforward implementation of dynamic load balancing is possible. A by-product of this work was a test client implementation that makes it easier to do a distributed load test of the MIST/C server. During this process we produced the largest configuration of MIST/C (or NEW) clients ever tested, confirming that the server will support over 200 clients in a session with no performance degradation, using a typical server platform. We intend to include the dynamic load balancing feature as a standard MIST/C deployment option in the future.

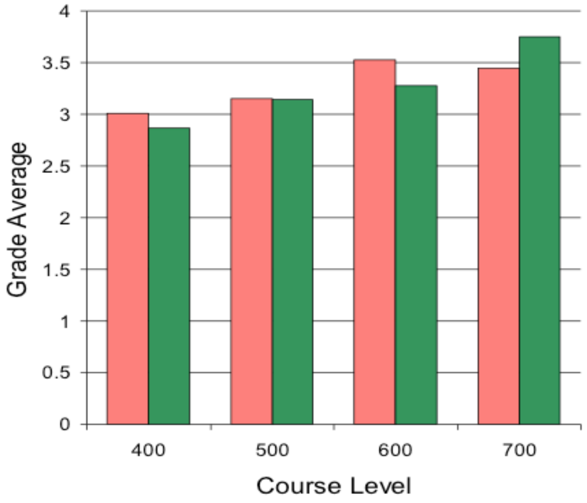
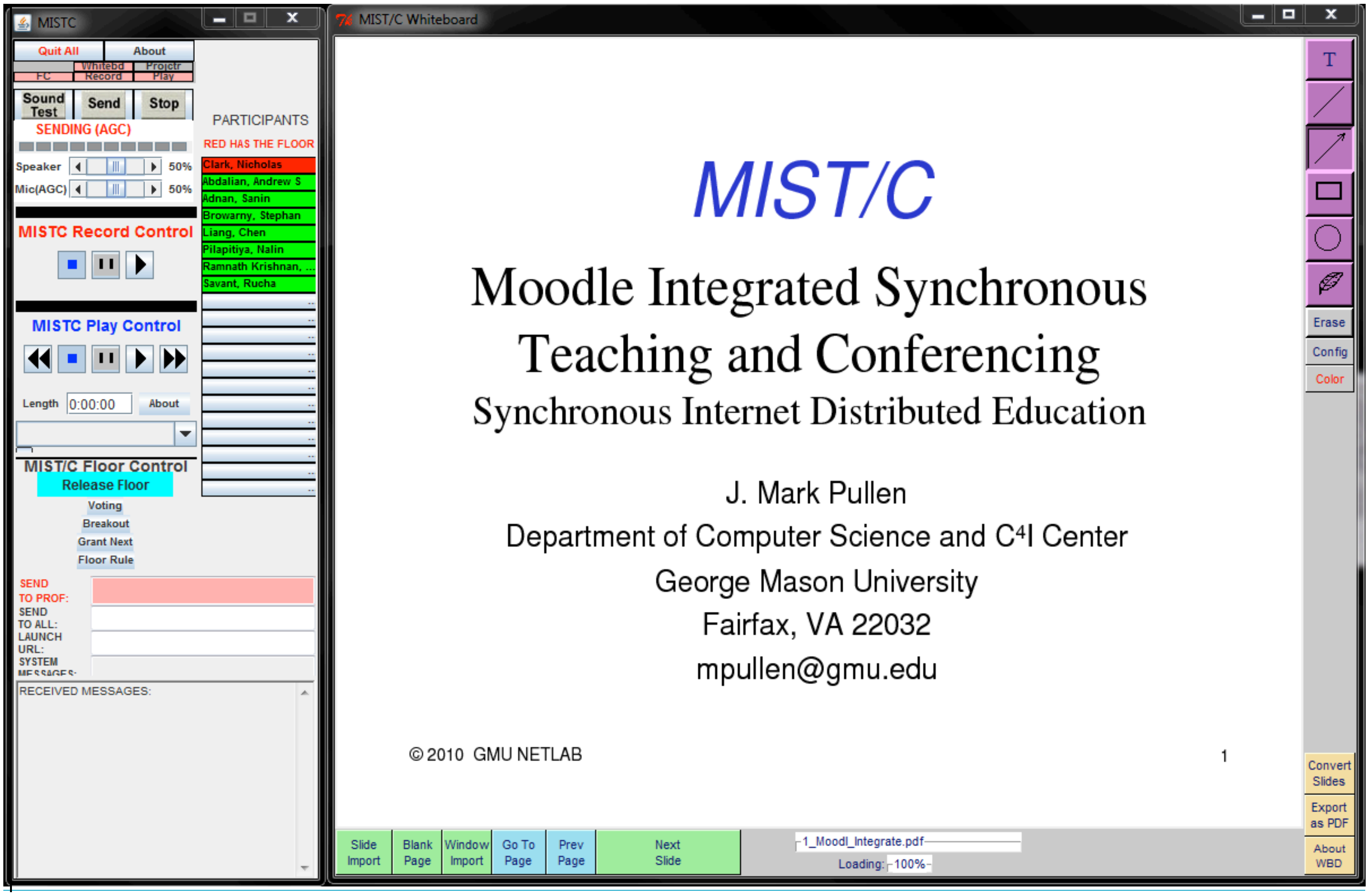


Figure 3. GMU CS grade comparisons
(pink: classroom; green: online)

7. Conclusion

MIST/C is a newly available open-source software system that provides a synchronous teaching and learning capability for the Moodle community, with the features of our older system NEW plus many enhancements. MIST/C is available free and open source and is easy to add to the Moodle environment. Early tests indicate that it has great promise to enrich the scope of free Internet education software to support synchronous and asynchronous DE in a single environment. More information is available at http://netlab.gmu.edu/MISTC.



MIST/C

Moodle Integrated Synchronous Teaching and Conferencing

Synchronous Internet Distributed Education

J. Mark Pullen
Department of Computer Science and C^{4}I Center
George Mason University
Fairfax, VA 22032
mpullen@gmu.edu

© 2010 GMU NETLAB 1

Figure 4. MIST/C User Interface.

References

[1] Hrastinski, S. Asynchronous and Synchronous E-Learning, *EDUCAUSE Quarterly,* 31(4), 2008

[2] H. Latchman, C. Salzmann, D. Gillet and J. Kim, Learning On Demand - A Hybrid Synchronous /Asynchronous Approach, *IEEE Transactions on Education* 44(2), May 2001 208-225

[3] Pullen, J., The Internet lecture: converging teaching and technology, *ACM Special Interest Group on Computer Science Education (SIGCSE) Bulletin* 32 (3) (Sep. 2000), 101-104

[4] Pullen, J.M. and J. Chen, Distributed Application Launching for High Quality Graphics in Synchronous Distance Education, *ACM Special Interest Group on Computer Science Education (SIGCSE) Bulletin,* 40(3), pp 204-208, 2008

[5] Snow, C., Pullen, J. and McAndrews, P. (2005), An Open-Source Web-Based System for Synchronous Distance Education, *IEEE Transactions on Education 48*, 4 (Nov. 2005), 705-712

[6] Pullen, J. and C. Snow, Integrating Synchronous and Asynchronous Internet Distributed Education for Maximum Effectiveness, Educational and Information Technologies (2007)3, pp 137-148, Springer, New York NY

[7] Sclater, N., Large-Scale Open Source E-Learning Systems at the Open University UK, (Research Bulletin, Issue 12). Boulder, CO: EDUCAUSE Center for Applied Research, 2008, available from http://www.educause.edu/ecar

[8] Moodle Site Statistics, http://moodle.org/stats, as of August 2010

[9] Snow, C., Pullen, J. and McAndrews, P., An Open-Source Web-Based System for Synchronous Distance Education, *IEEE Transactions on Education* 48(4) (Nov. 2005), 705-712

[10] Heffner, Kevin *et al.*, Modeling and Simulation Group 048 Technical Activity Final Report, NATO Research and Technology Office, Paris, France, 2010

[11] Moodle Open Source Software Project Developer Documentation, http://docs.moodle.org/en/Development: Developer_documentation, as of January 2011

[12] MIST/C website, http://netlab.gmu.edu/MISTC

[13] International Competition of Non-commercial Software Systems, Tools and Products for Web-Based Education, *IASTED Conference on Web-Based Education 2005*, Grindlewald, Switzerland, February 2005

[14] Moodle webpage, http://moodle.org

[15] MIST/C Server Downloads, http://netlab.gmu.edu/MISTC/server_vm.php

[16] SCORM Best Practices Guide for Content Developers, Carnegie Mellon Learning Systems Architecture Lab, 2004

[17] Clark, N., Pullen, J., McAndrews, P. and Snow, C., Moodle Integrated Synchronous Teaching/Conferencing: A Free Open-Source Synchronous Capability for Moodle, *Annual Conference on Distance Teaching and Learning 2010*, University of Wisconsin, Madison, WI, August 2010

[18] Pullen, J. and Clark, N., Moodle-Integrated Open Source Synchronous Teaching, *ACM Information Technology in Computer Science Education 2011*," Darmstadt, Germany, June 2011

[19] Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2, as of December 2010

[20] Pullen, J., Simon, R. and McAndrews, P., An Online Graduate Computer Science Program Delivered Via Simulteaching, *Advanced Technology for Learning* 2 (3), 148-155 (2005), ACTA Press, Calgary AB

[21] Pullen, J., Scaling Up a Distance Education Program in Computer Science, *ACM Special Interest Group on Computer Science Education (SIGCSE) Bulletin,* Vol 38 No. 3, pp 33-37, 2006

[22] Russell, T., *The No Significant Difference Phenomenon: A Comparative Research Annotated Bibliography on Technology for Distance Education (5th ed.)*, IDECC, 2001