

# Moodle-Integrated Open Source Synchronous Teaching

J. Mark Pullen  
Department of Computer Science  
George Mason University  
Fairfax, VA 22030  
+1.703.993.1538

mpullen@netlab.gmu.edu

Nicholas K. Clark  
C4I Center  
George Mason University  
Fairfax, VA 22030  
+1.703.993.1743

nclark1@netlab.gmu.edu

## ABSTRACT

It is well understood that the two prevalent approaches to distance education, synchronous and asynchronous, can be combined to good effect. However, existing open source software generally is separated into these two categories. This paper introduces an open source capability that combines the popular Moodle asynchronous learning management system with a new synchronous online teaching/conferencing system called MIST/C. This combination is supporting online delivery of the Master's programs in our Computer Science Department effectively. This paper describes the design and implementation of MIST/C and our experience using it to support Computer Science MS programs.

## Categories and Subject Descriptors

K.3.1 [Computing Milieux]: Computer Uses in Education – distance learning

## General Terms

Management, Performance, Human Factors

## Keywords

Internet distance education, synchronous, asynchronous

## 1. MOTIVATION

Distance Education (DE) often is lumped into two major categories: synchronous DE, where student and instructor communicate electronically in real time, and 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 complementary and can enable better effect together than either achieves individually [1]. However, the open source educational software community has been slow to produce systems combining these capabilities.

We share the observation of [1] that synchronous and asynchronous modes are more effective when combined. Therefore, when we discovered the software integration facilities in the Modular Object-Oriented Dynamic Learning Environment (Moodle), we were motivated to rework our previous product into the Moodle environment. Moodle is widely recognized as a high-quality Learning Management System (LMS) supporting asynchronous DE [2]. This paper is about the result: Moodle Integrated Synchronous Teaching/Conferencing (MIST/C) [3].

## 2. DISCUSSION

MIST/C provides audio, video, and whiteboard interfaces, a floor control/chat panel, a recorder, and a playback unit, all 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. It runs on Windows, Linux and MacOSX platforms. The following additional open-source features have been provided as part of Moodle integration:

- Auto-reconnect, so that ongoing class is not disrupted in case of network problems
- Server-side recording, to provide a backup to client recording
- Integrated control panel for all functions, with full student names
- Whiteboard functions for window capture and PDF slide format

We have used MIST/C extensively to teach Master's level Computer Science courses, blending classroom participation with on-line attendance. Both lecture and seminar styles have worked well in this mode, including cases where students made their project presentations remotely. A total of 24 courses taught by our Computer Science department have been presented. Student and faculty feedback has been positive, and has been used to improve MIST/C further.

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 and determined that a straightforward implementation of dynamic load balancing is possible. A byproduct 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 MIST/C 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.

## 3. REFERENCES

- [1] Hrastinski, S. Asynchronous and Synchronous E-Learning, *EDUCAUSE Quarterly*, vol. 31, no. 4, Oct-Dec 2008
- [2] Moodle Modules and Plugins, <http://moodle.org/mod/data/view.php?id=6009>
- [3] MIST/C website, <http://netlab.gmu.edu/MISTC>