Integrating Marratech video conferencing with the Moodle course management system

Mustafa Paksoy, et al.
Swarthmore College, Swarthmore, PA
mpaksoy1@swarthmore.edu

Abstract

Video conferencing is an integral part of online collaborative teaching, however most course management systems lack this key piece of functionality. Over this summer we integrated the powerful Marratech video conferencing suite with Moodle, a user friendly open source course management system. Our final product tightly integrates an application running on the LAMP web platform with another on Java Enterprise Edition. It provides a simple interface that adheres to Moodle's look and feel, and still allows the full power of Marratech to be harnessed. Throughout this project we followed a test-driven incremental software engineering methodology.

Summary

Course Management Systems (CMS) provide a wide array of collaborative tools that allow many components of a course to be conducted online. However, many CMS's lack video conferencing capabilities that are integral to a full online collaborative teaching environment. Working for the National Institute for Technology and Liberal Education (NITLE), we developed an application that tightly integrates a popular CMS with a powerful video conferencing suite. The resulting application provides to the user the best of both worlds.

Moodle is an open source CMS that puts great emphasis on simplicity and user friendliness. Moodle runs on the open source LAMP (Linux Apache MySQL Php) web platform. Moodle provides extensibility in the form of course activity modules. The Marratech video conferencing suite provides secure and versatile video conferencing. Marratech clients can run on Windows, Mac OS X and several popular linux distributions. The Marratech manager (the server) runs on a Java Enterprise Edition (JavaEE). The manager has an Application Programming Interface (API) that allows the provisioning of video conferencing rooms.

Our software development team consisted of ten undergraduate students from ten different liberal arts colleges. For the first three weeks of the nine week project our team was instructed by faculty members from several colleges on the core technologies relevant to our system. Our team then designed and implemented the application on its own over the course of six weeks. In order to minimize the amount of time lost designing our application we followed an incremental and test driven methodology called Extreme Programming. Our methodology allowed us easily adapt the resulting product to our client's (NITLE) changing requests.

Our final application has two main components: a course activity module for Moodle written in Php and several Java servlets written for Marratech. The Java servlets act as an interface to the Marratech API for our Moodle module. This module makes requests to the servlets over HTTP and seamlessly manages things like provisioning of rooms, creation of user accounts, and setting access privileges.

To the end user, our Moodle module simply appears as another activity that can be scheduled within a course. The user enters scheduling information and lists outside guests to invite. Our module then checks for conflicts base on the limitations of the Marratech license the user is using. When the time of the meeting comes the meeting is activated and users can enter the meeting from within Moodle.

The administrator of the Moodle site can customize the behavior of the module. The administrator can set, amongst other things, how strictly our system enforces meeting durations, the contents of email messages sent, the bandwith that each person in a meeting can use. The administrator can also sync with the Marratech manager at the click of a button.

The resulting application seamlessly integrates two separate systems running on completely different platforms. Through this our application combines the best aspects of the two systems it brings together. In short, it brings Marratech's power to bear while preserving the simplicity and user friendliness of Moodle.