

Molly Open Source Online Lecture System

Ronald P. Vullo, Ph.D., Natallia Ivaniuk, M.S.I.T., and Sean Boyle, M.S.I.T.

Golisano College of Computing and Information Sciences

Rochester Institute of Technology

Rochester, New York 14623

Abstract – *For over 13 years, one of the authors has been developing and using a Web-based presentation tool for in-class and distance education. Recent advances in Web technology have made it possible to re-implement the system and incorporate the ability for educators to easily add a video track and synchronize their slides to it. We also added shared permissions multi-user editing capability and several other features. This new version of a time-tested Web-based lecture slide system provides distance learners a classroom-like experience via the Web and makes it easy for educators to develop, edit, share, and reuse presentations.*

Keywords: e-Learning, Distance Education, Video, Open Source, Web

1 Introduction

For over 13 years, one of the authors (Vullo) has been developing and using a Web-based presentation tool as part of the open source Molly Website development system. Although used primarily in the classroom, it was conceived of originally as a distance education tool, and has been used in the delivery of at least one complete distance education course. In the early days of the system, distance lectures were created with an integrated video of the lecturer that drove and synchronized the slides. This was accomplished via QuickTime's HREF Track mechanism, combined with JavaScript. While it worked, it was fussy and required considerable expertise, time, and effort to author such lectures. Security enhancements to JavaScript eventually broke the system and a second technique — again driven by a special QuickTime track — was used. While more reliable, it was still a complex and tedious process to author a lecture. With the release of the HTML5 standard came the incorporation of the <video> tag. With video now part of the Document Object Model (DOM), it became possible to control both the

video and slides from JavaScript. More importantly, because the control codes and time codes no longer needed to be embedded in the video itself, it became possible to create a simple Web-based authoring environment. This finally made it possible to fulfill the original vision for the Molly system's distance learning component and was the genesis of this project. This paper describes the re-development and expansion of this lecture delivery system.

2 Project Overview

The Molly online lecture system emulates a real in-class environment and face-to-face communication with a professor by providing video lectures combined with automatic slide switching functionality. The main audiences for the system are students and faculty members. These user's groups have different business goals and needs that they wish to accomplish using our system.

During the design phase of the project, the requirements and expectations of the main intended audience were collected and analyzed. Students are interested in viewing any class materials online that will help them study their curriculum. Faculty members would like to use the system to share information with remote students as well as to make in-class presentations. The result of this analysis is presented in a form of use case diagrams.

There are two main actors in the system: Viewers and Editors. A Viewer is a user who is interested in viewing lecture content. It could be a remote student who is taking online class, as well as a professor who is presenting in class. An Editor is a user who is responsible for available lecture content and material (most typically the lecturer or

professor). For these audiences, the system has two key components: a presentation portal and a management portal. The presentation portal is intended for the viewing audience, while the management portal is for content administration and management.

Figure 1 (below) shows a use case diagram for Viewers. Viewers would like the system to support two presentation modes:

- In-class presentation: for the in-class environment, the presentations are displayed without any video content and with manual slide switching.
- Online presentation: this mode includes video with automatic slide switching. Available lecture content includes slides and any additional notes from the professor. For easy navigation between slides, the user must have the ability to select the next and previous slides, as well as random access to any of the other slides. In addition, students can see other people’s comments, suggestions and notes related to the slide content, and can post their own questions and comments for the whole community of users. In order to leave a comment and/or participate in forum discussions, a user must register themselves in the system. As far as the video, users have the ability to play, pause and seek. A print version is available containing all slide content and an additional notes section.

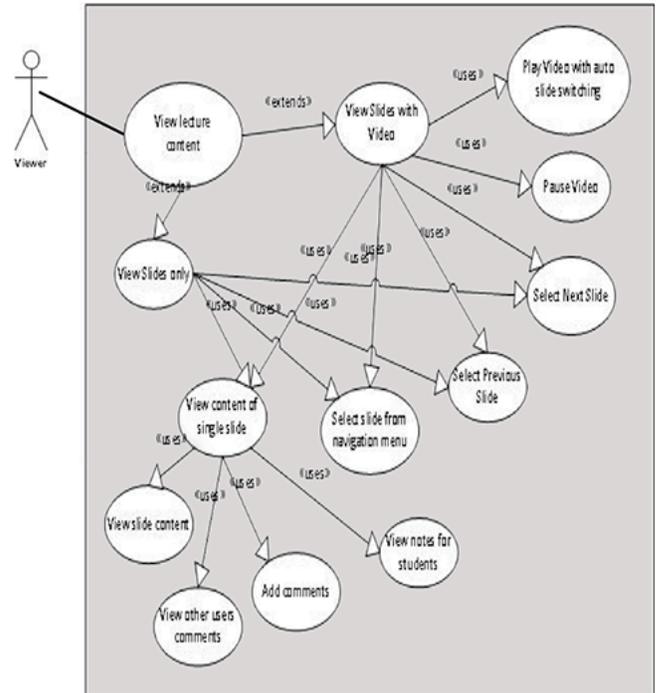


Figure 1: Use case diagram for viewers

Figure 2 (below) shows a use case diagram for Editors. Unlike Viewers, Editors are responsible for creating and managing lectures with all material required by the class curriculum. First, they need the ability to create lectures. The implemented solution provides two different ways of creating the context: either create a new lecture de novo or copy the content from an already existing lecture. The second approach is mainly suitable for classes with similar content; for example, classes that have versions taught to both graduate and undergraduate students. Editors must be able to see the list of all lectures that they have access to. They must be able to delete lectures, make lectures available online or take lectures offline.

The lecture content management functionality supports the capability to create, edit, delete, or reorder slides. Each lecture slide has an extra section for additional comments and notes regarding the slide content. Student notes are public and are available online for all viewers, while lecture notes are only available for Editors. The system provides the ability to upload pre-recorded video for the class with the video poster. It allows configuring automatic slide switching based on play position within the video content. For additional

security, the owner of the lectures can grant and revoke Editor rights to other users of the system.

current browsers: Firefox, IE, Safari, Opera and Chrome.

Business Server Layer

This layer encapsulates the main business rules and services. This layer enforces required parameter validation, data dependencies and data integrity. For developing the service layer the server-side scripting language PHP5 and Molly middleware (molly.rit.edu) were used.

Data Layer

The data layer is responsible for data storage and interactions with persistent data. It combines both data sets and the database management system. A MySQL database was used as a backend for the current implemented solution, however Molly uses a database abstraction layer allowing other databases such as Oracle and PostgreSQL.

4 Technologies and Tools

The system is implemented using the latest Web technologies such as HTML5, CSS3, JavaScript and AJAX libraries with a MySQL backend. The server-side component had been built using PHP and MAML.

Hypertext Markup Language (HTML5, DOM)

HTML5 is a new version of HTML that is rapidly penetrating the Web market. HTML5 is a markup language developed for presenting content for web pages.

HTML5 was first presented in 2004 by the Web Hypertext Application Technology Working Group. In 2008, the first working draft of the specification was published. Since then, many browsers have started supporting HTML5 features.

HTML5 significantly improves standardization for browsers' behavior and presentation across different platforms. Compared to older HTML versions, it offers new features for easier Web application development and more sophisticated form handling. HTML5 was the driving reason for this new version of the Molly system due to its integrated support for graphics, video and audio. It

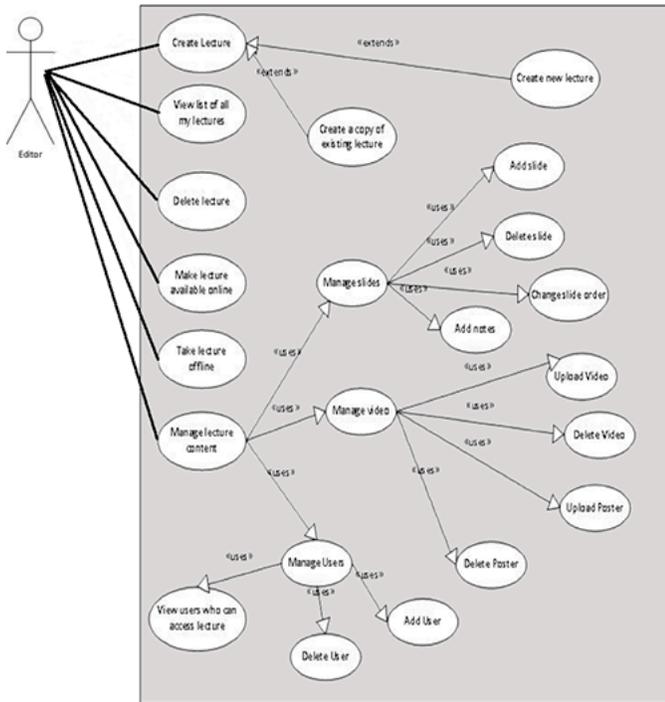


Figure 2: Use Case diagram for Editors

3 Application Architecture

The system has been developed using a standard three-tier architecture. In this model, the user interface, functional business logic, data storage and access are developed as independent components. This approach provides benefits such as reusability, flexibility, maintainability, and scalability.

Presentation Layer

This layer presents data to the user (Graphical User Interface). For the current implementation the latest web client-side technology, such as HTML5, CSS3, JavaScript and AJAX were used. The developed system is targeted to support the most common

provides APIs for embedding and controlling audio and video content. The new Canvas element provides support for 2D graphics animation without use of the Flash or Silverlight plug-ins.

Cascading Style Sheets (CSS)

Cascading Style Sheets are a set of rules that define how HTML elements are displayed and positioned on a web page. CSS styles can be defined in-line with HTML elements or imported from a separate style sheet. We used separate additional CSS files for this system, in addition to Molly's built in CSS, to simplify changing and configuring layout and styles of the page. As a result, in one of our future enhancements, users of the system will be able to select their own "skins" for the presentation of their slides.

Cascading Style Sheets were introduced 13 years ago, and are now ubiquitous on the Web. Style rules have significantly evolved over the years. The latest version, CSS3, offers very powerful features to make Web pages look cleaner and more sophisticated, such as rounded corners, multi-column layouts, borders for images, etc. CSS also allows slides to be formatted for printing automatically.

Cross-browser incompatibility in supporting different CSS styles, which leads to display differences, was a challenge for our design and implementation. In some cases, different attributes were used to support different browsers.

JavaScript

JavaScript is a lightweight, client-side scripting language. It is embedded in Web browsers in order to provide enhanced user interfaces and dynamic Web pages. JavaScript provides objects and classes to control the browser and its Document Object Model (DOM). It allows Web developers to place elements in an HTML form and respond to user events such as mouse clicks, form input and page navigation.

AJAX

The acronym AJAX stands for Asynchronous JavaScript and XML. It represents a set of web

technologies used to create dynamic web pages by enabling asynchronous communication between the client and server. Using AJAX, the client makes asynchronous calls to the server to send and/or retrieve data without refreshing the whole Web page. It significantly reduces the user's wait time and provides a wider and richer range of possibilities for user interactions.

PHP

PHP (PHP Hypertext Preprocessor) is an open source server-side scripting language used to develop rich Web applications. PHP code can be embedded directly into the HTML page. It runs on every common Web server platform, including Linux, Windows and Mac OS. Commonly, PHP is used as an Apache module, written in C, so it executes quickly and makes efficient use of system resources. PHP provides a rich set of libraries and APIs for graphics, database management, XML support, and much more. Because PHP is open source software, there are many sample solutions and considerable documentation available online. Its community provides reliable and quick support.

MAML

MAML (Molly Active Markup Language) is an XML markup language, developed by students of the Rochester Institute of Technology under the lead and guidance of Professor Ronald Vullo. The idea behind MAML is to create a generic library of controls to help abstract complex server-side logic, including database access, and help developers focus on client-side functionality.

XML Molly tags are included in HTML mark-up. Upon page load, the Molly system parses included tags and replaces them with built-in functionality written in PHP. For this system, we relied on built-in user management functionality, such as login control, and database access from the MAML library.

JQuery Library

There are many JavaScript libraries available for Web developers. Designing any Web site, we need to carefully consider the choice of additional libraries. We must preserve a balance between

features that the library offers and its impact to the page size and performance. JQuery is a JavaScript open source library (developed by a student of Dr. Vullo's) that provides better interaction between HTML and JavaScript by simplifying existing DOM APIs and providing cross-browser transparency. JQuery was first released at the beginning of 2006. Today, the lead Web development companies, such as Google, Amazon, IBM, and Microsoft, are all using JQuery for Web applications. According to BuildWith.com, more than 50% of the top Web sites are using jQuery libraries. JQuery has a rich library of methods for AJAX development, which were used in developing our system. This library significantly simplified drag and drop functionality for slide reordering. We also took advantage of its rich API for displaying popups and navigation menus with animated sliding effects. Also, JQuery provides a minimized version for its code that helps reduce its impact on our pages' load times.

KineticJS

For implementing slides and video synchronization, we decided to use the KineticJS library. During our search for the most suitable library, we considered some of the most common HTML5 libraries, including Fabric, Paper, Easel and Kinetic.

The *Fabric* library is mostly focused on vector graphics tools. It provides an easy API for manipulating objects in the Canvas. It supports simple objects like circles, rectangles and polygons, as well as more complex shapes. It allows one to dynamically move, scale, rotate and group elements. The size of the library is 78-148kB.

Paper is an open source vector graphics scripting framework. It offers a Document Object Model for easy object access and manipulation, and a great deal of powerful functionality to create and work with vector graphics. However, there is no mention in its documentation of support for multi-touch devices. The library size is 219kB.

Easel is more suitable for building HTML5 interactive and animated graphics. It would be a good choice for a game project. It has nice and easy

examples and documentation, as well as built-in support for touch devices. The library size is 65 kB. It seems that this library is a good choice for developers with Flash/ActionScript background.

KineticJS is a relatively new HTML5 library. However, thanks to its good documentation and support, it is quickly gaining popularity of the community. It is advertised to be very fast due to multiple canvases for speed. It allows one to draw shapes onto the stage, add event listeners to them, and move, scale and rotate them independently. The base code has good unit test coverage, which adds some confidence about its quality.

After some consideration, it was decided to go with KineticJS because it has all features needed for our purposes. It has good documentation and nice tutorials. It has very good event support, including support for mobile events and multiple browsers. The library is only 67kB. In addition, it shows very good performance.

TinyMCE is an open source JavaScript/HTML editor. It provides ability to convert HTML elements including text areas to editor instances. This editor offers a set of built in core HTML formatting tools, such as different font styles and decorations, formatted ordered and unordered lists, style alignments, etc. In addition to core functionality, there are many plugins available for TinyMCE to support additional features like image uploads. In our project, the JBIImage plugin for TinyMCE was used to allow image upload.

The TinyMCE Editor supports multiple configuration options that allow customizing and controlling its behavior. TinyMCE was already integrated into Molly and so was a natural choice.

File Structure

Figure 3, below, indicates the file structure for our solution. The root folder of the solution contains the main Web pages. Since the solution is built on the top of the Molly framework, all Web pages have *.maml file extensions. These pages are processed by the MAML parser and all MAML tags are replaced with the results of built-in PHP code.

The *commands* folder contains PHP files with main server side logic and database access. Web pages are making AJAX calls to these PHP files to retrieve data or to perform operations on the data. These files are also responsible for data validation and error handling.

The *includes* folder contains files with code snippets that are shared between more than one page. For example, the login control in title, page headers. These files are added to the page using Molly's `<maml:include>` tag.

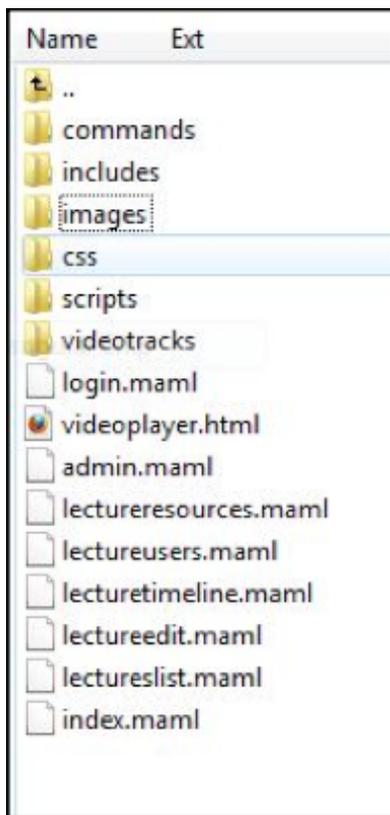


Figure 3: Project File Structure

The *scripts* folder contains JavaScript files. It has our custom scripts as well as third party libraries such as kinetic.js and jQuery scripts. In addition to such scripts, this folder also contains the TinyMCE controls with all required plugins.

The *css* folder contains files with CSS styling. JQuery CSS files are also located in this folder. Each Web page has its own style file with styles

specific to this page. The standard styles used across all pages are combined in the *ipad.css* file. The MAML controls are using their own styling located in the *decor* folder of the framework.

The *images* folder contains images for Web pages and controls not included in Molly framework. In addition, any images uploaded by users for lecture slides are stored in this folder.

The *videotracks* folder is where lecture authors store their videos and poster images. Video and images for each lecture are located in their own folders inside *videotracks*. To preserve folders' unique names, lecture IDs are used as folder names.

5 Future Enhancements

The following features are being considered as part of ongoing development:

- Translation support for multiple languages
- Mobile device support
- Touch-screen support
- Lecture content versioning and the ability to revert to previous versions
- Automatic change-tracking and recording of change history
- Ability to customize the look and feel of individual lectures by providing different layout templates and skins (currently only possible for the entire site)

6 Conclusion

Our university, like many others, is looking to increase the use of technology to facilitate distance learning. This goal is all the more important as we continue to expand our international campuses around the world. The project described in this paper is one important piece of an online education strategy that has the potential to reach a vast number of students and does so with modest technology requirements.

7 References

B. Means, Y. Toyama, R. Mupphy, M. Bakia, K. Jones “Evaluation of Evidence-Based Practices in Online Learning: A Meta Analysis and Review of Online Learning Studies”.

Ed.gov. US. Department of Education. 2010. Web. 1 Sep 2012.
<http://www2.ed.gov/rschstat/eval/tech/evidence-based-practices/finalreport.pdf>

D. Driscoll. Review of Moodle Course Management System. Purdue University. N.d. Web. 1 Sep 2012.
http://www.bgsu.edu/cconline/Web_2_0_Reviews/Moodle_Driscoll/index.html

“Learning Management System (LMS) Review. Summary of Findings”. Dec 2009. Duke University. Web. 1 Sep 2012.
<http://elearning.duke.edu/wp-content/uploads/2010/02/LMS-Review-Report.pdf>

“Premier Learning Unit Case Study”. RMD Studio. 2011. Web. 15 Sep 2012.
<http://www.rmdstudio.com/archived-projects/40-premier-learning-unit-case-study>

Sakai. Longsight, Sakai Commercial Affiliate. N.d. Web. 10 Sep 2012.
<http://www.sakaiproject.org/about-sakai>

<http://support.apple.com/kb/TA25532>

http://www.w3schools.com/html/html5_video.asp

Vullo, Ronald P., Ph.D.; Molly: Simplifying Development of Complex Web Apps, invited presentation to the Rochester Joint Chapters of the IEEE Computer and Computational Intelligence Society, Rochester, New York (June, 2012)

Vullo, Ronald P., Ph.D., Bogaard, Daniel S.; Better than HTML Web: XML for Program-ming-Free Dynamically Generated Web sites, WWW@10 (2004)

Vullo, Ronald P., Ph.D., Christopher A. Egert, Ph.D., Daniel S. Bogaard; Molly: Bringing Back Simplicity to Web Site Development and Web Research. White Paper, 2006

Vullo, R.P., et al. (October, 2000). Telemedicine initiatives in International Outreach at St. Jude Children's Research Hospital. Presentation at the American and European Associations for Cancer Education Joint Meeting.