Developing and Deploying Online Courses with JCourse

Willie Wheeler, David Danks, Joseph Ramsey,
Richard Scheines, Joel Smith, Andrew Thompson
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, U.S.A.
wwheeler@andrew.cmu.edu

Abstract: JCourse, created at Carnegie Mellon University, is a Java- and XML-based system for developing and deploying web-based courses. On the development side, JCourse allows content providers to work more independently of web designers than has been previously possible. On the deployment side, JCourse provides basic (albeit incomplete) support for the IMS Question and Test Interoperability v1.0 specification (www.imsproject.org), which allows questions and tests from one compliant system to be reused in other compliant systems. Because Java and XML are platform-independent, JCourse runs on Windows and Unix (including Linux). JCourse was used to create a web-based course on Causal and Statistical Reasoning (www.phil.cmu.edu/projects/csr) that has been successfully used by about 150 students at the University of California, San Diego, the University of Pittsburgh, and Carnegie Mellon University.

JCourse supports the development and deployment of web-based courses. Development support includes instructional and assessment pages. Deployment support includes an assessment engine, a user interface for student and instructor access to student performance data, and security-related access control.

Without JCourse, instructional content for online courses is typically created as HTML web pages by the content author. The instructor uses his or her favorite text or HTML editor to create the HTML file, and then FTPs it to a web server so the students can access it.

This approach works well when instructional materials do not require professional design from a visual design perspective, but otherwise it falters. (Consider, for example, an online course intended to be packaged and distributed as a commercial product.) Because content authors are subject experts, not visual designers, the course development effort requires a separate visual designer role.

HTML presents several major problems in such a situation. First, since HTML encodes both content and presentation, content authors and visual designers cannot work independently, which reduces the number of design iterations. Second, content authors produce non-HTML versions of the content (e.g., Microsoft Word), which leads to synchronization issues. Third, HTML does not provide a mechanism for keeping multiple pages consistent in appearance. Finally, the course developers may want to give the course a completely new look from time to time. For large courses this is difficult with HTML, as each page must be individually updated.

JCourse solves the above by separating content and presentation into different files. The content files use a simple XML (eXtensible Markup Language) format appropriate for the content author. The presentation files, or stylesheets, are based on XSL (eXtensible Stylesheet Language), a sophisticated language that visual designers can use to define page appearance. The XSL stylesheets are applied to the XML content files to obtain web-ready HTML files, which can then be viewed using any web browser without special plug-ins. The transformation process is called XSLT (XSL Transformations). XML, XSL, and XSLT are all W3C technologies (www.w3c.org). Using JCourse, authors and designers can work more independently, allowing rapid design-test-revise iterations. Since the content format is simple, there is no need for the multiple copies that become out of sync. Having a stylesheet enforces a consistent look across pages. Finally, changing the look of the entire site is simply a matter of changing the stylesheet (or small number of stylesheets) responsible for that look.

For assessment development, JCourse provides basic support for the IMS QTI specification, which defines a portable XML-based format for the creation of questions and tests. As the format is complex, JCourse includes an assessment authoring application, implemented using Java Swing. The interface allows the author to create IMS-compliant questions and tests, including single- or multiple-response choice questions, and fill-in-the-blank

questions. JCourse also provides extension functionality not directly supported by the IMS QTI specification, including support for tables and Shockwave movies. Questions and tests created in JCourse can be used in any IMS-compliant system, as long as they use only nonextension functionality. Similarly, questions and tests created in IMS-compliant systems can be used in JCourse.

JCourse can also be used for course deployment. JCourse includes facilities, in various stages of completion, for course management (e.g., maintaining class rosters), student account management (e.g., viewing one's assessment scores), delivering instructional pages and assessments, tracking student performance, and security-related access control. JCourse runs as a Java web application on any web server that supports the Java 2.2 Servlet web application specification. (This includes most major web servers; see www.java.sun.com.) A single JCourse instance can host multiple courses and multiple instances per course.

The JCourse assessment engine performs three major functions: assessment delivery, response processing, and results storage. The user requests an assessment by clicking on a hyperlink on the web page. The hyperlink points at the URL where the assessment engine resides, and the query string includes an assessment ID so the engine knows which assessment to deliver. The assessment engine then loads the assessment, and generates HTML dynamically for that assessment, using JavaServer Pages (JSP). It sends the HTML back to the client browser, which displays it. At this point the student answers the question, presses the submit button, and sends the results back to the assessment engine. The assessment engine evaluates the results according to the assessment specification as created by the authoring tool. It then writes the performance data to the student performance database, and finally returns feedback to the student.

The JCourse gradebook gives course instructors simple web access to the student performance data. The instructor can view the score for any given student/assessment pair, as well as summary statistics for given students or for given assessments.

Deployment options are flexible. Course instances can be configured to have their own question and test repositories, or they can share them. Sharing saves disk space and simplifies management by avoiding the need to synchronize updates, while having individual repositories allows customization. Similarly, course instances may share student performance repositories, or have their own. In the case of performance repositories, a JDBC-based abstraction layer sits between the repository and the application proper, and allows any JDBC-compliant database (e.g., Access, SQLServer, Oracle, Sybase, mSQL, and MySQL) to be used to persist student data.

JCourse was used to create a web-based course on Causal and Statistical Reasoning (www.phil.cmu.edu/projects/csr) that has been successfully used by about 150 students at the University of California, San Diego, the University of Pittsburgh, and Carnegie Mellon University. Approximately four people, all working part-time on the project, produced the bulk of the course in under a year. The material includes dozens of Shockwave simulations, almost 500 questions tracked and recorded in the JCourse database, and a large virtual laboratory in Java, all delivered in approximately 18 content modules.

Classes at UCSD and at Pitt are delivering the same material to students with traditional lecture/recitation format, and with JCourse. On the first midterm, students using JCourse scored 6 percentage points higher at UCSD, and 3 points higher at Pitt. Although these groups were self-selected, a similar experiment in the spring of 2000 found the selection effect to vary between 2 and 4 percentage points. Students using JCourse to learn the fundamentals of experimental design and causal inference are faring as well as their peers in lecture.

Future directions for JCourse include the following:

- More complete support for the IMS QTI specification, including support for additional question types and support for question selection and sequencing when it is specified.
- Performance-based (as opposed to security-related) access control. For example, students cannot progress to later course sections until they have demonstrated mastery of earlier sections.
- External API to the student performance database so performance data can be collected for exercises implemented as applets, Shockwave movies, etc. We already have several such applets (Causality Lab, Carnegie Proof Tutor, D-Separation Tutor, Set Builder), but currently they provide their own persistence capabilities. We want JCourse to provide at least general support, such as saving scores and timestamps.
- We will investigate the possibility of open sourcing the project.