Creating and Using Learning Objects With Open Source Tools

Jan Cheetham
Division of Information Technology
University of Wisconsin-Madison

Les Howles
Division of Information Technology
University of Wisconsin-Madison

Christine Lupton
Division of Information Technology
University of Wisconsin-Madison

This session will define the concept of learning objects and show examples of learning objects created with freely available open-source authoring tools. It will demonstrate how these objects can be incorporated into a variety of Web-based e-learning scenarios by drawing from a wide range of faculty exemplars.

The focus is on the development of reusable learning objects and how they can be created using freely available open-source software tools. The session discusses the concept of learning objects and shows examples of learning objects created for a variety of sample courses. It describes the concept and philosophy underlying open-source authoring for creating learning objects and outlines the attributes of open-source authoring tools (compared to commercial and shareware applications). For those interested in the open-source initiative, the session provides information on where and how to obtain these tools.

What Is a Learning Object?

In simple terms, a learning object is a unit of instruction, delivered digitally. David A. Wiley, in “The Instructional Use of Learning Objects,” defines them as “any digital resource that can be reused to support learning” Wiley continues: “The main idea of ‘learning objects’ is to break educational content down into small chunks that can be reused in various learning environments, in the spirit of object-oriented programming.” (http://www.reusability.org/read/)

The Educause Learning Initiative (formerly NLII) describes learning objects as “modular digital resources, uniquely identified and metatagged, that can be used to support learning.” (http://www.educause.edu/eli)

The Wisconsin Online Resource Center describes learning objects as Web-based, self-contained chunks of learning that are small enough to be embedded in a learning activity, lesson, unit or course. They are flexible, portable, and adaptable, and can be used and reused in multiple learning environments and across disciplines. (http://www.wise-online.com/)

For the purposes of this paper, learning objects must meet the following criteria:

1. reusable
2. self contained
3. provide instructional content
4. address a specific learning goal
5. adhere to a valid instructional method/strategy
Learning objects can be built using a variety of authoring tools, including the open source tools listed at the end of this paper. Most authoring tools consist of an easy to use interface that allows the instructor to simply add his or her content into a template that contains pre-defined formats for organizing, displaying, and annotating content.

“Open Source” — What Does It Mean and Why Should You Care?

The beauty of open-source software development can be summed up in a familiar phrase: “Have it your way.” Under open source, the licensing, source code, and development are all open. Anyone can view and freely modify everything related to “open-source” software. All source code that makes up open-source software and its documentation are easily available at no charge.

What’s more, if you use or modify the software, “We’re okay with that.” Anyone who modifies the software is asked to share his or her work with others, thereby enriching the entire development and user community. Those who modify open-source software can even package and sell their version, if they so choose.

Some standards govern open-source software; in fact, some software that is publicly available might not meet those standards. The Open Source Definition (OSD), for example, sets some limits on use, modification and redistribution of open-source software. OSD has also provided a framework for various licensing arrangements for open-source software.

Commercial applications are much more (often totally) restrictive in their access to underlying source code. Commercial software licenses prohibit users from modifying the applications or limit their ability to do so.

Shareware is somewhat less restricted. Shareware is usually distributed free of charge, with the author requesting payment of a small fee from those who use the software regularly. Anyone can copy and distribute the software, and any new user is expected to pay the fee to the author. But shareware is copyrighted, and users are not free to modify the software and cannot sell it as their own.

Two examples of open-source development initiatives in higher education are Sakai and SCORM.

The Sakai Project (http://www.sakaiproject.org) is a nationwide collaborative initiative to build online educational tools in an open-source environment. Founded by the University of Michigan, Indiana University, MIT, Stanford and two open-source software development groups, Sakai enables the higher-education community to take advantage of work being done at peer institutions, primarily in the development of course management software. Because Sakai is open source, innovators can play an important role in developing instructional tools. This leads to “cross-pollination” and a level of participation by developers and eventual users that is difficult, if not impossible, with more traditional vendor-supplied software.

SCORM (http://www.adlnet.org/), the site of Advanced Distributed Learning (ADL), a collaborative effort to modernize structured learning) — the Sharable Content Object Reference Model — defines a Web-based framework for aggregating, describing, and sequencing learning objects and for defining communication and data to be tracked for learning objects. It provides a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reuse of Web-based learning content. One outcome of the collaborative SCORM effort is coordination of emerging technologies and commercial and public implementations. It also provides recommendations to the vendor community.
SCORM allows for scalable, reusable, sharable course content; discoverable learning content (enables interoperable repositories); the ability to find and move entire courses; vendor support; and the development of adaptive learning systems that can assemble content to meet the learner’s needs “on the fly.”

Why should anyone be interested in developing learning materials in an open-source environment? Open-source development might seem to be nothing more than altruism — a lot of work shared widely, merely for the good of the community. But developers can benefit from the collaboration. As the Open Source Initiative (OSI; www.opensource.org) says: “When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing. We in the open source community have learned that this rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of bits.”

What about intellectual property rights in an open-source setting? The real intellectual property value is in the content. The tools might be open, but the content need not be. In any event, open-source developers are only filling gaps in the marketplace, and commercial products will certainly not disappear.

**Three Open-Source Tools Built at UW-Madison and Freely Available**

The Engage project at the University of Wisconsin-Madison has developed three open-source authoring tools that can be used to create reusable learning objects. These tools — ConceptTutor, QuizImage, and DynaLesson — are freely available for download at [http://engage.doit.wisc.edu/tools/](http://engage.doit.wisc.edu/tools/).

1. **ConceptTutor Authoring Tool** — This authoring tool uses an intuitive Macromedia Flash environment to lead authors through a very simple process of creating learning objects. ConceptTutor includes elements common to most authoring environments, including such functions as Save, Preview, Publish, and Edit.

   The learning objects, themselves called ConceptTutors, are “just-in-time” HTML windows that open when needed by the learner from existing Web pages. They teach a “concept,” a class or group of things or ideas that share a unique set of attributes and are referred to by a common name or symbol.

   We like to think of ConceptTutors as glossaries on steroids. They support various media types (images, audio, and animations), and their design is based on best practices for learning concepts. Effective concept learning is accomplished by including the concept definition, examples and non-examples, contextual information, and quizzing, in an interactive tab format.

2. **QuizImage Authoring Tool** — The QuizImage authoring software runs in Java and creates an HTML file that contains the learning activity. This HTML page, itself called a QuizImage, is a learning object that uses images for quizzing and can be hyperlinked to existing Web pages. The authoring tool is easy to use, and instructors can create a simple activity from an image in less than five minutes. With a little more time, authors can add detailed feedback and questions to build a richer learning experience.

   QuizImage enables instructors to create interactive images, which can be effective for teaching about ideas contained in visual representations. These interactive learning objects can teach ideas in a visual context and give students a chance to test their understanding and
recall of ideas. The images are useful for helping students learn visual skills such as reading maps and graphs or finding visual metaphors in works of art.

QuizImages help learners in two ways:

A. Integration of words and pictures. Picture-word combinations such as those in QuizImages help learners build connections between ideas (Mayer et al., 1995) and retain information longer than when presented as pictures or words alone (Haber and Myers, 1982).

B. Self-assessment. QuizImages work as self-tests that let students measure their understanding of content.

3. DynaLesson Authoring Tool — DynaLesson authoring software enables instructors to create pedagogically rich online course materials in the form of fully independent, Web-delivered lessons. These lessons provide dynamic activities that include one or more of the following:

- Multimedia elements (video, audio)
- Welcome and exit pages clearly outlining learning goals and next steps
- Activities, such as multiple choice, matching, short answer, and essay
- Multimedia Annotator, an interactive media player providing help
- The ability to stand alone or be used within course management systems.

For those interested in creating reusable learning objects using tools created in the open source community context, please see the following references for more information.

**Open Source Resources for Creating Learning Objects**

ATutor
http://www.atutor.ca/
Course management system and associated tools.

Engage
http://engage.doit.wisc.edu
Authoring tools for creating interactive images (QuizImage, concept-based lessons (ConceptTutor), and multi-media lessons (DynaLesson).

Moodle
http://moodle.org
Course management system and associated tools.

Pachyderm
http://www.nmc.org/pachyderm/index.shtml
Authoring tool for creating multimedia learning experiences.

Project Pad
Software for organizing materials from digital repositories and for annotating and integrating them into web-based projects.
Sakai Project
http://www.sakaiproject.org/
Course management system and associated tools.

University Educational Technology Services – Georgia State University
http://www2.gsu.edu/~wwwets/instructionalsupport/learningobjects/finding.html
Templates for creating learning objects that are timelines, simple case studies, and game-like activities.

Presenters’ Contact Information

Jan Cheetham
Division of Information Technology
University of Wisconsin-Madison
1301 University Ave.
Madison, WI  53715
Phone: 608 262-2947
E-mail: cheetham@wisc.edu

Les Howles
Division of Information Technology
University of Wisconsin-Madison
1301 University Ave.
Madison, WI  53715
Phone: 608-265-5045
E-mail: llhowles@doit.wisc.edu

Christine Lupton
Division of Information Technology
University of Wisconsin-Madison
1301 University Ave.
Madison, WI  53715
Phone: 608-265-8948
E-mail: christine.lupton@doit.wisc.edu