Open–Source Online Knowledge Portals for Education

Phillip Olla
Madonna University, USA

Rod Crider
Wayne Economic Development Council, USA

INTRODUCTION

The open-source community has created a broad suite of educational and e-learning course management systems (CMS) referred to as educational knowledge portals (EKP). An EKP is a software system designed to aid instructors in the management of online educational courses for their students, especially by helping teachers and learners with course administration. These systems make it possible for a course designer to present to students, through a single, consistent, and intuitive interface, all the components required for a course of education or training.

The system can often track the learners' progress, which can be monitored by both teachers and learners. Components of these systems usually include templates for content pages, discussion forums, chat, quizzes, and exercises such as multiple-choice, true/false, and one-word-answer testing. New features in these systems include blogs and really simple syndication (RSS) technology. Services generally provided include access control, provision of e-learning content, communication tools, and administration of user groups. They might also provide functions like threaded discussions, chat, grade books, course outlines, file sharing and digital content display. This chapter uses the term educational knowledge portal, however these e-learning systems are sometimes also called learning management systems (LMS), virtual learning environments (VLE), education via computer-mediated communication (CMC) or online education. They might also be called a managed learning environment (MLE), learning support system (LSS) or learning platform (LP).

The purpose of this chapter is to introduce the concept of open-source knowledge portals and highlight some of the benefits and risks associated with using these types of systems. This chapter will also explore some of the open-source systems that are currently available and successfully used by educational institutions.

OVERVIEW OF OPEN-SOURCE KNOWLEDGE PORTALS

Course management systems are now commonplace in higher education and faculty are becoming more sophisticated about the use of educational technology in teaching. Products like WebCT, Blackboard and eCollege provide comprehensive and integrated tool sets, but were designed for ease of use and not to meet the needs of more sophisticated users (Collier & Robson, 2002).

Most academic products on the market today were initially conceived as solutions for departments, or even single courses. Their underlying architectures did not anticipate the need to scale to many thousands of students and to smoothly integrate with student information, financial, human resources, and other academic computing systems. The price of the course management systems themselves is rising as vendors strive for profitability, and anecdotal evidence indicates that some campuses have written more lines of code to integrate their campus systems with a vendor's course management system than there are lines of code in the vendor's system itself. Massive customized in-house development brings with it not only cost but also an increased risk of a system failure that cannot be diagnosed or that cannot be fixed at a reasonable cost.

The open-source software movement offers the greatest opportunity in the creation and distribution of knowledge and information. Preservation of openness and sharing (at an educational level) is critical for the creation of a culture that values innovation, progress, experimentation and development.

Open-source software refers to computer software and the availability of its source code for use under an open-source license to study, change, and improve its design. In 1998, a group of individuals presented "open source" to re-label free software in order for such software to become more mainstream in the corporate world (Wikepedia, 2006). Open-source software generally allows anybody to make a new version of the software, port it to new operating systems and processor architectures, share it with others, or market it. The aim of open source is to let the product be more understandable, modifiable, 'duplicapable,' or simply accessible, while it is still marketable.

The open source definition presents an open-source philosophy, and further defines a boundary on the usage, modification and redistribution of open-source software. Software licenses grant rights to users which would otherwise be prohibited by copyright. These include rights on usage, modification and redistribution. While open source presents

a way to broadly make the sources of a product publicly accessible, the open-source licenses allow the authors to fine tune such access.

It is important to differentiate between the types of software licensing. The term "open-source software" is used by some people to mean more or less the same category as free software. It is not exactly the same class of software as open-source. However, the categorical differences are small: nearly all free software is open-source, and nearly all open-source software is free.

Benefits of the Open-Source Approach

The American Bar Association and Rasch (2006) report that there are many reasons why the open-source model has been successful and popular with developers, including the following:

- **Access to Source Code:** Documentation for commercial software products is often lacking on detail and out-of-date. This can be challenging to developers who try to write software programs that are designed to interoperate with or target other programs. Having access to source code enables the developer to understand the program at a deep level and to debug and optimize his or her own program at a level of efficiency and skill that is often not possible with programs available only in binary form.
- **Broad Rights:** The broad license grant, which allows licensees to use, modify, and redistribute open-source programs, is a major advantage of the typical open-source license. Typical commercial software products are distributed only in binary form and may not be modified. Often the documentation associated with commercial programs is not detailed enough to permit some kinds of "value added" programming that is possible for developers who have direct access to source code.
- **Encourages Software Re-Use:** Open-source software development allows programmers to cooperate freely with other programmers across time and distance with a minimum of legal frictions. As a result, open-source software development encourages software re-use.
- **Can Increase Code Quality and Security:** With closed source software, it is often difficult to evaluate the quality and security of the code. In addition, closed source software companies have an incentive to delay announcing security flaws or bugs in their product. Often this means that their customers do not learn of security flaws until weeks or months after the security exploit was known internally. Open-source software is potentially subject to scrutiny by many eyeballs. Therefore bugs, security flaws, and poor design can-

not hide for long, at least when the software has a community of programmers to support it. And since fixing the code does not depend on a single vendor, patches are often distributed much more rapidly than patches to closed source software.
- **Decreases Vendor Lock-In:** Businesses no longer have to be locked-in to a sole-source vendor. This reduces the need to constantly upgrade simply to maintain compatibility with others using the same software. Business data is also more "future-proof," since most open-source programs save text files in ANSI standard ASCII files, instead of proprietary binary formats.
- **Reduces Cost of Acquisition:** Most open-source software is available for a nominal cost, often the price of the media, or the time of the download. Reduced acquisition cost means that start-ups do not have to part with capital when they need it most. Established companies can try the software with minimal risks. If a company wants to develop a piece of software that is not proprietary, they can reduce the cost by collaborating with several companies on the same code base. Expensive per-seat license fees are also eliminated.
- **Increases Customizability:** Every business has unique needs or desires that can be addressed. Linux has been ported to everything from embedded microcontrollers to IBM mainframes. If there is a bug to be fixed, anyone can be hired to fix it. If two programs have interoperability problems, one or both can be modified to eliminate the incompatibility.
- **Community:** Having a common source code pool and the tools provided by the Internet creates an opportunity for extensive and speedy collaboration on development projects.

Risks Associated with Open Source

Open-source development models can also expose organizations to some disadvantages. Some of these include:

- **There is no Guarantee that Development Will Happen:** It may not be possible to know if a project will ever reach a usable stage, and even if it reaches it, it may die later if there is not enough interest. Especially when a project is started without strong backing from one or more companies, there is a significant initial gap when the source base is still immature and the development base is still being built. If it is not possible to get funding or enough programmers cooperating at this stage, the project just "dies," or perhaps slowly fades out.
- **There may be Significant Problems Connected to Intellectual Property:** It can be very difficult to know if some particular method to solve a software problem is patented, and so the community can be considered

Related Content

Supply Chain Management and Portal Technology

Scott Paquette (2007). *Encyclopedia of Portal Technologies and Applications (pp. 997-1001).*

www.irma-international.org/chapter/supply-chain-management-portal-technology/17999

The ABCs of Designing Campus Portals

Ali Jafari (2003). *Designing Portals: Opportunities and Challenges (pp. 7-27).*

www.irma-international.org/chapter/abcs-designing-campus-portals/8216

Lotus Workforce Management-

Jerh. O'Connor, Ronan Daltonand Don Naro (2010). *International Journal of Web Portals (pp. 22-36).*

www.irma-international.org/article/lotus-workforce-management/40316

Using the Building Blocks: Evolution of a Portal Suite

Joe Lamantia (2012). *Enhancing Enterprise and Service-Oriented Architectures with Advanced Web Portal Technologies (pp. 259-271).*

www.irma-international.org/chapter/using-building-blocks/63961

Analysis of Accessibility Initiatives Applied to the Web

Joselice Ferreira Lima, Gustavo Miranda Caran, Luiz Fernando R. Molinaroand Daniela Favarro Garrossini (2012). *International Journal of Web Portals (pp. 48-58).*

www.irma-international.org/article/analysis-accessibility-initiatives-applied-web/75531