

The Emergence of Open-Source Software in North America

Guohua Pan and Curtis J. Bonk

Volume 8, Number 3, November 2007

URI: <https://id.erudit.org/iderudit/1071864ar>

DOI: <https://doi.org/10.19173/irrodl.v8i3.496>

[See table of contents](#)

Publisher(s)

Athabasca University Press (AU Press)

ISSN

1492-3831 (digital)

[Explore this journal](#)

Cite this article

Pan, G. & Bonk, C. (2007). The Emergence of Open-Source Software in North America. *International Review of Research in Open and Distributed Learning*, 8(3), 1–17. <https://doi.org/10.19173/irrodl.v8i3.496>

Article abstract

Unlike conventional models of software development, the open source model is based on the collaborative efforts of users who are also co-developers of the software. Interest in open source software has grown exponentially in recent years. A Google search for the phrase open source in early 2005 returned 28.8 million webpage hits, while less than two years later that number had jumped to 376 million. This paper discusses the origin of the term open source and the key tenets of the open source software development model. In addition, it analyzes the merits and drawbacks of using this model and discusses the advantages and disadvantages of applying the model in higher education. Importantly, examples are provided of computer software, and course management systems in particular, developed using the open source model. Also included are brief analyses of the Linux operating system, and two open source course management systems, Sakai and Moodle, as well as the uPortal. A timeline of major open source projects of significance in North America is provided. The paper concludes with a discussion of the potential for applying the open source software development model to open and distance education.

Copyright (c) Guohua Pan and Curtis J. Bonk, 2007



This document is protected by copyright law. Use of the services of Érudit (including reproduction) is subject to its terms and conditions, which can be viewed online.

<https://apropos.erudit.org/en/users/policy-on-use/>

This article is disseminated and preserved by Érudit.

Érudit is a non-profit inter-university consortium of the Université de Montréal, Université Laval, and the Université du Québec à Montréal. Its mission is to promote and disseminate research.

<https://www.erudit.org/en/>

November – 2007

The Emergence of Open-Source Software in North America

Guohua Pan

MacEwan College, Canada

Curtis J. Bonk

Indiana University, USA

Abstract

Unlike conventional models of software development, the open source model is based on the collaborative efforts of users who are also co-developers of the software. Interest in open source software has grown exponentially in recent years. A *Google* search for the phrase open source in early 2005 returned 28.8 million webpage hits, while less than two years later that number had jumped to 376 million. This paper discusses the origin of the term open source and the key tenets of the open source software development model. In addition, it analyzes the merits and drawbacks of using this model and discusses the advantages and disadvantages of applying the model in higher education. Importantly, examples are provided of computer software, and course management systems in particular, developed using the open source model. Also included are brief analyses of the *Linux* operating system, and two open source course management systems, *Sakai* and *Moodle*, as well as the *uPortal*. A timeline of major open source projects of significance in North America is provided. The paper concludes with a discussion of the potential for applying the open source software development model to open and distance education.

Keywords: open source, Bazaar model, redistribution, untapped resources

Introduction

In recent years interest in open source has grown exponentially. In mid January, 2005 a *Google* search of the phrase open source returned approximately 28.8 million webpage hits (Kapor, 2005). That number jumped significantly to 376 million when a *Google* search was conducted on October 12, 2006. Given recently difficult monetary times, there is keen interest in higher education and corporate training related to the use of open source software. Some organizations and institutions are choosing such tools as podcasting, wikis, and blogs for their glamour and excitement, as well as open source tools and systems in more mundane areas such as financial systems, enrollment services, and learning management systems.

The open source model of software development, which increased in visibility after Raymond (1997) published his essay “The Cathedral and the Bazaar,” provides an approach to software development that is different from the conventional model of software development. As

indicated, it promises great potential for the public at large, and for higher education in particular, to use software extensively for effective learning and instruction. Linux has been frequently cited as one of the most successful examples of open source software. A number of colleges and universities in North America have been developing course management software and associated services using an open source development model.

Not surprisingly, there are many questions raised concerning the open source movement. For instance, what is open source? How does it work? And what are the implications of the open source development model for learning in general, open and distance education in particular? In seeking to explore some of the answers to those questions, a literature review on open source and learning was conducted. We conducted an extensive Web search on open source and open source learning management systems. We visited the websites of open source learning management systems that we felt were of high importance in open source movement. We interviewed some prominent figures of the open source movement and administrators of some open source management systems for information not available on their websites. We analyzed three examples of course management software developed by colleges and universities in North America using the open source model.

Open Source Software Development

The label open source code first drew media attention at a strategy session held in Palo Alto, California on February 3, 1998, after Netscape's announcement of the release of Navigator's source code for its Web browser Mozilla (Wikipedia, 2006). Instead of the word 'free' that appears confrontational to business world, Christine Peterson, one of the influential figures of open source movement at the session came up with the pragmatic and business friendly label 'open source.' Also at the session were people in leadership roles in the open source movement, including Todd Anderson, Larry Augustin, John Hall, Sam Ockman, and Eric S. Raymond (Eric Raymond, personal communication, January 31, 2006). The term open source has since been widely adopted to mean any computer software program whose source code is free to its licensed users for use, modification, and redistribution. In effect, open source refers to a product, usually an original computer software program that is "of or relating to source code that is available to the public" either partially or in whole (Answers.com, 2005a). This definition of open source deals only with the juridical category of software licenses, that is, with the terms of distribution and redistribution of the software.

More interesting than pinning down a definition of it however, is the development process surrounding open source software. An open source software product differs from commercial software in that an open source software product, tool, or system is "created by a development community rather than a single vendor" and that the users are individuals working independently or affiliated with participating organizations to rewrite the source code (Answers.com, 2005a). Under the open source development model, the licensed users are potential developers who contribute to the development of the open source software code by rewriting the original code. The modified version is revised and then released frequently for further modification. As such, the "source code of open source software is free and available to anyone who would like to use it or modify it for their own purposes" (Answers.com, 2005b). One interesting and unique aspect about open source software is that an organization does not have to wait for a new release of the software for new functions and features. Instead, it can add desired features to the existing program itself. Moreover, the organization can redistribute copies of either the original or the modified program (Wheeler, 2006).

In summary, Johnson (2005) paraphrases the following criteria for open source definition:

1. *Free Redistribution* – Copies of the software can be made at no cost.
2. *Source Code* – The source code must be distributed with the original work, as well as all derived works.
3. *Derived Works* – Modifications are allowed; however, it is not required that the derived work be subject to the same license terms as the original work.
4. *Integrity of the Author's Source Code* – Modifications to the original work may be restricted only if the distribution of patches is allowed. Derived works may be required to carry a different name or version number from the original software.
5. *No Discrimination Against Persons or Groups* – Discrimination against any person or group of persons is not allowed.
6. *No Discrimination Against Fields of Endeavor* – Restrictions preventing use of the software by a certain business or area of research are not allowed.
7. *Distribution of License* – Any terms should apply automatically without written authorization.
8. *License Must Not Be Specific to a Product* – Rights attached to a program must not depend on that program being part of a specific software distribution.
9. *License Must Not Contaminate Other Software* – Restrictions on other software distributed with the licensed software are not allowed.

There are various advantages of open source. One of the advantages is that it motivates innovation – original source code provides a base for the receivers to begin with while frequent discussion of code improvement results in idea exchanges essential to innovation. In an open source world, original ideas illuminate the receivers' world and provide the spark and motivation to others in the community or conversation to do better.

Another frequently cited advantage is that open source makes available the talent of the world (Johnson, 2005; Kapur, 2005; Robles, 2005; Room 17, 2005; Wheeler, 2006). It is a way to lever those untapped resources because, as Bill Joy, one of the founders of Sun Microsystems stated, "most of the smart people in the world work somewhere else." Since a new software release is frequently conducted immediately after any significant changes are made and anyone can participate in the development and modification of the code, open source is believed to be "fundamentally a more efficient as well as democratic way of developing software" (Kapur, 2005, p. 72).

A third advantage is that open source reduces the cost and helps to create a sustainable economy wherein co-developers' participation in code development is free. This third advantage may be of particular significance to higher education institutions and will be discussed later in this paper.

What makes open source may also break it, however. In open source, the quality of the product is often at risk because anyone can download source code, work on it, and redistribute the 'finished' product. Since there are no guarantees that this new code is of high quality and without problems, the finished product is thus at the mercy of the co-developers' reputation. Equally problematic, the open source product is often left unattended to or forgotten once the original developer decides to no longer fund it and offloads this product as open source for the world community (e.g., e-education for Jones Knowledge, Inc.). As a toddler in the information age, this open source movement is analogous to the old wild west where there were few fences and spotty law enforcement with lots of cowboys on the move looking for greener pastures (Kapur, 2005). A few

such cowboys rise in fame, but most are faint memories or sudden tragedies of some rival gang. Dead links act like grave stones reminding us of their short existence and untimely deaths.

Another disadvantage is the hidden cost. In many cases there is no technical infrastructure to support some critical requirement or function in the open source product. Additional modifications, and, in effect, more money and other resources, are therefore needed not only to keep the project running but to make others aware of it and enlist their support and services.

The Bazaar Model

It has been widely accepted that the visibility of the open source development model has been substantially raised since Raymond (1997) put forward the concept of the Bazaar development model (Kapor, 2005; Kivekäs, 2005; Morin, 1998; Robles, 2005; Room 17, 2005; Wheeler, 2006). In his essay “The Cathedral and the Bazaar,” Raymond (1997) postulates a new model of software development. He calls this new model or approach the “bazaar model” to distinguish it from the traditional “cathedral model” of software development. In the cathedral model, the software development process is centralized and there is a relatively strong control over who can submit patches to the code and how they are integrated, as well as a rigorous plan for code releases. Under this model, source code is available with each software release, but code developed between releases is restricted to an exclusive group of developers (Wikipedia, 2006).

In contrast to the cathedral model, the central thesis of the bazaar model is that the development of software is distributed and transparent. According to Raymond (1997), users of the operating system are potential developers with different agendas and approaches. Raymond notes that, in effect, the source code of the prototype software is open and released as early as possible to attract co-developers, even though it may have limited functionalities. Besides, the software is released whenever significant changes are made such that the product evolves in an incremental way, enabling co-developers and users to modify and debug it. Other features of the bazaar model include the co-existence of several versions; for example, a production version that is stable, and a development version that is unstable but supplies all the latest functionality. Last but not least, a dynamic decision making structure exists in either a formal or an informal fashion to make strategic decisions (Kivekäs, 2005; Morin, 1998; Raymond, 1997; Robles, 2005).

A criticism of the bazaar model is its oversimplification – any open source project appears the same and will be completed, whether the project is a large scale one with millions of lines of code or a more modest one with thousands of lines (Bezroukov, 2005a). The reality is that the bazaar model is difficult to reproduce and even tougher to predict where it is headed. As a result, there is continued uncertainty as to whether it will ultimately lead to success or failure. In part, this tension arises due to the fact that extremely few projects will excite sufficient interest for software developers to continue working beyond their normal 8-12 hour work (days) (Robles, 2005; Room 17, 2005).

One of the bazaar model’s tenets, according to Raymond (1997), is that “given enough eyeballs, all bugs are shallow” (§ 1). This is particularly problematic in the case of kernel development, because the number of highly eager and available co-developers is just one of the necessary factors for successfully debugging a complex system. In terms of *Linux*, a kernel consists of the core code of the operating system which is responsible for providing [secure](#) access to the machine’s [hardware](#) and to various computer processes.

Linux and Other Open Source Projects

Linux, a free computer operating system, has been repeatedly cited as one of the earliest open source projects, and, more importantly, one of the most successful open source models. A key reason for this success is that it presently is the only viable alternative to *Microsoft's WindowsNT* (Bezroukov, 2005a, 2005b; Johnson, 2005; Machado & Thompson, 2005 ; Raymond, 1997; Robles, 2005). Linus Torvalds, from Finland, is the creator of *Linux*. *Linux Kernel 1.0*, the heart of all *Linux* systems, was developed and released under the [GNU General Public License](#) in 1994. Its source code is freely available to everyone. Amazingly, based on this kernel, hundreds of companies and organizations and an equal number of individuals have released their own versions of the *Linux* operating system (Linux Online, 2005).

Distributed code development is one of the most frequently cited advantages in the development of *Linux*. Based on the *Linux* kernel, the code is re-written by the co-developers who are also licensed users across the world. This distributed development process allows the *Linux* to tap a talent pool from across the planet with minimal cost while addressing the problems of individual users. Of course, it is the Internet which actually makes this globally-based and locally-focused development process possible.

Another reason for the success of *Linux* is its large user base. A key reason for the widespread use of *Linux* is because it can be downloaded for free. And for people, institutions, or organizations unable to download it because of bandwidth, Internet accessibility, or other related issues, it can be obtained from a retailer such as the CD-ROM Shop in Canada for less than CDN \$20. The large number of users has meant that there are a plethora of needs, which forces the development of *Linux* to become more sophisticated. At the same time, *Linux* was PC friendly from the beginning, enabling its user base to grow along with the increasing popularity of the PC.

Another critical factor contributing to the success of *Linux* is the frequent releases of new *Linux* versions and the parallel debugging by a huge army of co-developers. This creative development process is akin to a Delphi approach or effect wherein the new code is based on the average opinion of many equally expert observers and is deemed to be more reliable than the opinion of one randomly-chosen individual expert observer. There is always an able person or team who amidst the mass of talented co-developers can fix a tricky problem (Raymond, 1997). Duplication is minimized in *Linux* since fixes or patches are propagated and fed-back as quickly as possible so that any inefficiency resulting from the duplication of parallel debugging is limited, if not eliminated (Raymond, 1997). At the same time, parallel development in *Linux* is feasible because over half of the code in *Linux* developed by co-developers correspond to device drivers, which typically are relatively independent of the code kernel (Godfrey & Tu, 2001). Of course, more critical fixes occur when co-developers are involved in developing a significant change to the code corresponding to the heart of kernel.

Other challenges in using *Linux* include the necessary technical skills required for a user to run *Linux* system at its optimum as well as the maintenance costs and security measures to protect the computer from malicious attempts (Pfaffenberger, 2000).

Open Source in Higher Education

Higher education institutions, and in particular colleges and universities, are places wherein innovation and the free exchange of ideas are fostered (Hilton, 2005). Universities play a central

role in the creation of new knowledge as well as the transmission, distribution, and ultimate application of it within the economy and society as a whole (Hilton, 2005).

Along with the massive use of information technology (IT) in the classroom, higher education is facing severe financial constraints because an institution's base budget is not only for licensing fees, but also for training and support required for application software (Rhodes, 1999; Wheeler, 2006). For example, in a survey of IT executives from 52 colleges and universities on IT-related issues, the executives responded that pricing and scalability were the most important factors when buying course management systems for their institutions and they all expressed their dissatisfaction with the cost of course management systems from vendors such as *WebCT* and *Blackboard* (Hartrey & Fallon, 2005).

An option for an institution to achieve sustainable economics is to limit changes related to application software so as to reduce the institution's total spending on IT; fewer changes to track and announce, training events, and orientation sessions. Yet, colleges and universities are noted for extensive networks of widely dispersed and talented individuals who are motivated to develop innovative tools, resources, and pedagogical methods, especially those online. As a result, software applications that are constraining or offer limited opportunities for experimentation or creativity are unlikely to be applauded within higher education institutions (Rhodes, 1999).

- Universities and colleges are constantly facing the plethora of challenges and tensions related to utilizing resources that are financially sustainable and, perhaps more importantly, also have underlying standards, while simultaneously developing creative products or software application programs that push the frontiers of the possible (Rhodes, 1999; Wheeler, 2004). A new business model is needed in higher education, and in all educational settings, to address these dual challenges – to utilize the sound products that exist while allocating resources to develop custom products where what is available is insufficient or inferior. The success of *Linux*, and, more importantly, the model of open source application development on which *Linux* is developed, appears to be a highly valuable solution for colleges and universities (Johnson, 2005; Raymond, 1997; Wheeler, 2006).
- Following the open source development model of *Linux* and backed by like-minded software vendors and IT foundations, higher education institutions in North America initiated various open source projects that focus mostly on course management systems and related systems and tools. Some popular and widely known open source tools used in higher education are detailed in Table 1.

Analogous to a 'greenhouse' for growing open source projects, colleges and universities have begun to embrace the open source development model resulting in many new and exciting open source projects and products (Abel, 2005; Wheeler, 2004). One striking feature of those open source projects, as Table 1 illustrates, is that they are mostly open source course management systems, student electronic portfolios, or student information systems that seem specially developed for those in higher education with limited applicability to corporate and government training settings (Abel, 2005). In addition, most well known open source projects in higher education have been undertaken collaboratively by several institutions. Perhaps such development efforts not only utilize the hard work, experiences, and thought power of these institutions but also the ethical models of higher education where strong, collaborative communities of sharing and idea exchange are the norm, instead of hiding one's ideas from competitors (Abel, 2005).

Table 1. The open-source offerings in North America

Project	Institution(s)	URL
<i>Apache</i>	Apache Software Foundation	http://www.apache.org/
<i>Atutor</i>	University of Toronto	http://www.atutor.ca/index.php
<i>Bazaar</i>	Athabasca University	http://klaatu.pc.athabascau.ca/cgi-bin/b7/main.pl?rid=1
<i>Chandler OSAF</i>	Open Source Applications Foundation	http://www.osafoundation.org/
<i>DSPACE</i>	MIT Libraries and Hewlett-Packard (HP)	http://www.dspace.org/
<i>FEDORA</i>	Cornell and Virginia	http://www.fedora.info
<i>Globus</i>	Argonne National Laboratory, Chicago, Southern California Information Sciences Institute, Edinburgh, Swedish Royal Institute of Technology, and Northern Illinois	http://globus.org
<i>Haystack</i>	MIT	http://haystack.lcs.mit.edu
<i>LionShare</i>	Penn State , MIT Open Knowledge Initiative, Simon Fraser , and the Internet2 P2P Working Group	http://lionshare.its.psu.edu/main/
<i>LRN</i>		
<i>Moodle</i>	Moodle partners	http://www.moodle.org
<i>OKI (Open Knowledge Initiative)</i>	MIT	http://web.mit.edu/oki/
<i>opensourceCMS</i>	The Open Source Collective, Inc.	https://www.opensourcehost.com/mos452/
<i>OSPI (Open Source Portfolio Initiative)</i>	Carnegie Foundation for the advancement of teaching, Delaware, Minnesota, Rhode Island, and the r-smart group,	http://www.osportfolio.org/
<i>PKI</i>	Dartmouth College	http://www.dartmouth.edu/~pkilab
<i>Sakai</i>	Indiana , Michigan, MIT, Stanford	http://www.sakaiproject.org/
<i>Shibboleth</i>	Brown, Carnegie Mellon, Columbia, The Ohio State, Internet2 and the NSF Middleware Initiative	http://shibboleth.internet2.edu/
<i>Uportal</i>	JA-SIG members Princeton, UBC, Boston College, Delaware, and Sun Microsystems	http://www.uportal.org/
<i>Visual Understanding Environment (VUE)</i>	Tufts University , the Mellon Foundation	http://vue.tccs.tufts.edu/
<i>WeBWork</i>	University of Rochester	http://webwork.math.rochester.edu/

<i>Whiteboard</i>	SourceForge	http://whiteboard.sourceforge.net/
<i>Wikipedia</i>	Wikipedia, the free encyclopedia	http://www.wikipedia.org/

Note: Table 1 is derived from 3waynet Inc. and the Commonwealth of Learning, 2003; Abel, 2005; Coppola & Neelley, 2005; Farrell, 2003; Wheeler, 2004; Wikipedia, 2005.

Of the tens of open source projects outlined in Table 1, *uPortal* is often cited as a successful model (Coppola & Neelley, 2004; Wheeler, 2004; Yanosky, Harris, & Zastrocky, 2003). *uPortal* is a free, sharable “common portal reference framework,” which includes “ a set of technical specifications, and software” (Gleason, 2001, p.15).

The framework provides a J2EE portal server (container) and well-defined interfaces that will permit individual institutions to customize the institutional portal by plugging in components in a well-defined and usable manner. The portal specification provides single, sign-on plug-and-play, providing both ease of use (removing the need to sign-on each time an application is accessed) and the ability to implement single sign-on in a way appropriate to the situation. The portal specification defines interfaces for the content suppliers (publishers), allowing for smooth integration of channels and applications. (Gleason, 2001, p.15)

The participating institutions of *uPortal* are both consumers and developers in that they can publish the content through their institutional portal while consuming the content of interest published on *uPortal*. An institutional portal may be described as a virtual university, or a one-stop service station consisting of “applications that provide a single, intuitive, and personalized gateway to access and to integrate campus-specific information and applications with unstructured data from on and off campus” (Gleason, 2001, p. 14). Typical applications delivered via the portal include campus administrative systems, library information systems, and learning management systems. *uPortal* is built with Java, XML, JSP and J2EE. There are over 90 participating institutions across the world, including universities from North America: Athabasca University, University of British Columbia, University of Calgary, Columbia University, Cornell University, Dalhousie University, Duke, Memorial, Princeton, University of Saskatchewan, and Yale. Also participating are those in Bristol, Edinburgh, Nottingham, Oslo, and Stockholm within Europe, as well as universities in Hong Kong (i.e., University of Science and Technology), Japan (i.e., Nagoya), New Zealand (i.e., Auckland University of Technology), and Vietnam (i.e., Vietnam National) (*uPortal*, 2006).

Sakai is another popular open source example in higher education. The *Sakai Project* is a community source project to produce an open source collaboration and learning environment (CLE). The founding institutions and organizations in the *Sakai* project include Indiana University, MIT, Stanford University, the Open Knowledge Initiative (OKI), the University of Michigan, and the *uPortal* Consortium with the support of the Andrew W. Mellon Foundation and the Hewlett Foundation. Following the community source model, each participating institution contributes to the *Sakai* project by developing “the Sakai application framework and associated CMS tools and components that are designed to work together” (Sakai, 2005). Specifically, *Sakai* course management system incorporates the best features of the existing course management systems of CHEF developed by University of Michigan, CourseWork by Stanford, Oncourse by Indiana, and Stellar by MIT, as well as their development experiences. Using that experience, these four institutions have partnered their resources, code, and modules to build even more robust and valuable course management systems, testing and examination software, and other tools for universities around the world. Those applications aim to address the

common and unique needs of the participating institutions (Sakai, 2005). Recent data indicate that the number of participants is growing fast with 97 participating institutions in just two years (Sakai, 2006).

Moodle , an acronym for the Modular Object-Oriented Dynamic Learning Environment, is yet another popular open source software package designed to help educators create quality online courses (Moodle, 2005a; Wikipedia, 2005). Like other open source software, *Moodle* is free to download, use, modify, and even distribute and sell (under GNU General Public License), all with no license fee. According to information at the *Moodle* website, one of the main advantages of *Moodle* over other systems is its strong grounding in social constructionist pedagogy embedded in the features that “support role sharing, allow(ing) each participant to be a teacher as well as a learner” (Wikipedia, 2005). Instead of being the ultimate source of knowledge or sage, an instructor can role model the type of class culture that they seek to implement and attempt to personally connect with students while addressing their individual needs and experiences. In addition, the instructor in *Moodle* can moderate or facilitate discussions and activities in ways that help students negotiate ideas and socially interact to meet course learning goals (Coppola & Neelley, 2005; Moodle, 2005a; Wikipedia, 2005).

Moodle operates without modification on *FreeBSD*, *Linux*, *Mac OS X*, *NetWare*, *Unix*, *Windows*, and any other system that supports PHP , including most Web host providers. Data is stored in a single database: MySQL and PostgreSQL are best supported, but it can also be used with *Access*, *Interbase*, *ODBC*, and *Oracle* (Moodle, 2005b; Wikipedia, 2005) .

As of October 17, 2006, *Moodle* was available in 75 languages, with 1 7,095 sites from 160 countries that had, in fact, registered their *Moodle* installation (Moodle, 2006). It has been reported that the largest single *Moodle* site holds over 6,000 courses and enrolls over 45,000 students, and the *Moodle* installation at [Open University](#) of the UK [is built](#) for their 200,000 users. (Coppola & Neelley, 2005; Moodle, 2005a; Wikipedia, 2005).

Although there is increasing interest in research on open source (Carey & Gleason, 2006; Downes, 2006a; Hepburn & Buley, 2006; Huang, Dong, & Ge, 2006; Iiyoshi, Richardson, & McGrath, 2006; Quamen, 2006; Stephenson, 2006; Wiley, 2006), most of what we now know about open source is descriptive information about how it works and why it is an important trend. There is limited, if any, information about what implications it has for open and distance learning.

Milestones of, and Approach to, the Open Source Movement in North America

Long before open source became a label to denote collaborative efforts for computer software development in 1998, the socially diverse production model was exercised by engineers and computer scientists when they used open standards such as the Internet Request for Comments (RFC) in their development of the Internet in 1969. RFC consists of a series of notes, policies, and/ or protocols for the Internet (Internet Mail Standard, 2006; RFC.net, 2006; Wikipedia, 2006).

A further look into the approaches of the open source projects reveals that in North America, open source is more an individual initiative derived from personal needs with minimal government support. The free software movement upon which open source is built, for example,

was initiated by Richard Stallman, a computer programmer who quit his full time job at MIT to develop *GNU*, the first open source project. Another example is *Linux*. Linus Torvalds was a university computer science student when he released *Linux 0.01*, the first computer operating system licensed under *GNU* General Public License. *Linux* was the first open source software that became a viable competitor to proprietary software such as the *Microsoft Windows* system. As a third example, Martin Dougiamas from Western Australia spearheaded the development of the *Moodle* course management system when he became unhappy with the *WebCT* and other standard CMS products.

Table 2. Important Open Source Projects in North America

Time	Sponsor	Project	Significance
1969	Internet Society	<i>Request for Comments (RFC)</i>	The <i>de facto</i> exercise of first open source software development.
1983	Richard Stallman	<i>GNU project</i>	Start of free software movement, on which open source is built.
1991	Linus Torvalds	<i>Linux</i>	First viable open source operating system under <i>GNU</i> .
1995	Apache Software Foundation	<i>Apache Server HTTP</i>	Apache was the only viable open source alternative to the Netscape web server in 1995.
1995	Profs. Arnold Pizer and Michael Gage at the University of Rochester	<i>WeBWork</i>	Written in both Perl and LaTeX code, WeBWork individualizes problems, provides students with immediate feedback, encourages students to make multiple attempts, and provides instructors with real-time statistics for lesson plan customization.
1998	Mozilla Organization	<i>Mozilla Application Suite</i>	First Web browser to release its source code.
1999	Martin Dougiamas	<i>Moodle</i>	An open source eLearning platform that supports social constructivist epistemology.
2000	JA-SIG, the Java in Administration Special Interest Group, JA-SIG member institutions	<i>uPortal</i>	A set of Java classes and XML/XSL documents that can be customized to produce a portal for use on campus.
2000	Bertelsmann Foundation, the Heinz Nixdorf Foundation, the Sal. Oppenheim Foundation and the Department of Science and Research of the State of Northrhine-Westphalia	<i>ILIAS</i>	first <i>free software LMS</i> that has reached SCORM 1.2 Conformance Level LMS-RTE3, guaranteeing platform independent re-use of contents.
2001	Wikipedia, the free encyclopedia	<i>Wikipedia</i>	First multi-lingual Web-based free-content encyclopedia collaboratively written by

2001	Andrew W. Mellon Foundation, MIT, Stanford	<i>Open Knowledge Initiative (OKI)</i>	volunteers. First collaborative efforts by multiple universities to build and share components of course management systems, and to develop standards for such systems.
2001	Athabasca University	<i>Bazaar</i>	One of the first open source learning management systems in Canada capable of delivering courseware, portals or any of the other myriad types of web based projects.
2002	MIT	<i>OpenCourseWare</i>	University education becomes freely accessible to anyone anywhere any time.
2002	MIT Libraries and Hewlett-Packard	<i>DSPACE</i>	An open source software package providing the tools for the management of digital assets.
2002	OpenSourceHost	<i>opensourceCMS</i>	A content management system allowing users to add and delete content, change the way things look without fear of breaking anything.
2002	SourceForge	<i>Whiteboard</i>	Whiteboard supports multiple departments and courses; allows simple migration of courses to new semesters, grade storage, checking, and calculation; submitted assignment retrieval.
2002	University of Toronto, The Learning Disabilities Resource Community, Industry Canada, HRDC Canada, Canadians Gateway, Alberta Online Consortium, TMD, Oklahoma Department of Career and Technology Education, University of Bologna, American Academy of Ophthalmology, International Relations and Security Network, Labor Scarl, Eurisko, Norwegian Centre for Telemedicine, Fraser Health Authority	<i>ATutor</i>	First of its kind in Canada with accessibility to all user, including those with disabilities, and its adaptability, being able to configure the system to many different learning scenario
2004	Indiana , Michigan, MIT, OKI, Stanford, uPortal, Andrew W. Mellon Foundation	<i>Sakai Project</i>	An open source Collaboration and Learning Environment (CLE) software that requires

2004	Louvain Foundation	<i>ClarolineProject</i>	<p>purposeful community source efforts through participating members' explicitly defined roles, responsibilities, and funded commitments.</p> <p>A free online collaborative learning platform allowing a large variety of pedagogical setup including widening of traditional classroom and online collaborative learning, has been translated into 31 languages</p>
------	--------------------	-------------------------	---

The Open Source Development's Relation With, and Contribution to, Open and Distance Learning

The 'gift culture' and distributed development might be one of open source development's contributions to open and distance learning. Raymond (1997) notes that, in effect, the source code of the prototype software is open and freely available to the users who are potential co-developers, even though the source code only has limited functionalities. While the gift culture allows people to access source code freely, distributed development calls those same people, in one way or another, to participate in improving the prototype such that it becomes sophisticated (i.e., with more and stable functionalities). This is a give-and-take process that creates an obligation for people to give back when a gift is given; the values and beliefs of the giver may also be passed on to the recipients. It thus binds people together. As Weber (2004) notes, "the artifact being gifted is not just a functional widget but carries with it some of the giver" (p. 149). This gift culture and distributed development is further manifested in the Open Source Initiative's (2006) introduction to the idea of open source: "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."

This practice of gift culture and distributed development in open source can also be – and may have already been – practiced in open and distance learning. Participants in an online discussion, for example, can post their ideas on a given topic freely while reading peers' ideas on the same topic. The posting of such honest appraisals, supports, and advice is an example of social as well as cognitive presence wherein learners project themselves into a community of inquiry as real people (Rourke & Anderson, 2002). Their ideas, albeit incomplete and perhaps unsophisticated, function in the same manner as the source code with limited functionalities in open source. Through the same give-and-take practice, that is, by referring to peers' postings, commenting on peer postings, and being commented on by peers, what were once incomplete and shallow ideas now become increasingly complete and sophisticated. In addition to such volunteer contributions, however, motivational measures, such as reputation and/ or status within the system or community, may be utilized to encourage and reward students' contributions to the continued refinement of ideas, as what happened in open source software development. And a mechanism may be established such that people would not feel comfortable if they did not contribute, or contributed even a little. Enabled by the Net, this open source model provides "expanded opportunities for students to plunge ever deeper" into learning and affords "a near limitless means for students to grow their knowledge" (Anderson, 2004, ¶13).

The gift culture and distributed development in open source also shows that it is feasible and effective to build and consolidate an online community. As the evolution of *Linux* manifests into a community of ‘knowledge sharing,’ each person will be working on and improving the software. When this occurs, developers and co-developers can increasingly look at things from similar paradigms, and interact with each other and negotiate meanings out of “a common ground of interest and understanding” (Bonk & Cunningham, 1998, p. 31). This means that meaning must be accepted by the members in the community of practice before it is considered valid. Although each member has their own preference to, or belief in, each and everything, there is a set of important core understandings that members of the community share in common. These shared understandings consist of the norms, values, attitudes, beliefs, and paradigms, or in other words, the culture, in this community of practice. The culture, albeit implicit, thus helps build and consolidates subsequently a community of practice that is grounded on the same experience the members of this community of practice share, and a concomitant knowledge as how they should be able to perform (Cole, 1996; Fisher, Giaccardo, Eden, Sugimoto, & Ye 2005; Mahoney, 2004).

A community of practice is possible and feasible in open and distance learning too. The values of and beliefs in sharing and distributed development, or the culture, are promoted and reinforced while open and distance learners share their ideas with peers, comment on peers’ ideas, and defend their own. Typically, such transactions occur through asynchronous communication tools and are grounded in existing knowledge contexts (Anderson, 2004). Consciously or unconsciously, such open and distance learners use the language and act in ways that are acceptable to them. A culture, thus, begins to form that unites the learners and makes them feel an increasingly sense of belonging to each other.

At the same time, the expanded possibilities afforded by open source model can be overwhelming. It remains a challenge to maintain open and distance learners’ interest and commitment to the projects they are in because of the spontaneity feature of participation.

Conclusion

This paper reviewed the definitions and connotations of open source and the bazaar model of software development. In contrast to software development, acquisition, and use in the past, the two most striking features of the open source development model are distributed development of software and free redistribution of the software copies. As a result, *Linux*, an open source operating system, is seen as a viable alternative to *Microsoft’s Windows* system. Equally significant, the Internet appears to have been shifting from being a medium for information transmission and consumption to that of a platform through which content is created, shared, remixed, repurposed, and passed along by its participants to potential users (Downes, 2006b).

Like many open source advocates, we discuss *uPortal*, *Sakai*, and *Moodle*, three popular open source projects that are being undertaken by higher education institutions in North America in response to their dual challenges of developing sustainable economics and advancing innovation for application software and other related products in higher education. Given these open source trends of the past decade, and especially the past few years, there are many exciting opportunities for research and development within higher education across the human reaches of this particular planet and beyond.

The ‘gift culture’ and distributed development enable the success of open source software, such as *Linux*, and help to build and consolidate a community of practice of open source software developers. This gift culture and distributed development can, and may have already been,

mirrored in open and distance learning practice, while they may also help to build and reinforce a community of open and distance learners. The spontaneity and expanded possibilities of the open source model can also impact open and distance learners adversely.

References

- 3waynet Inc. and the Commonwealth of Learning. (2003, June). *COL LMS Open Source*. Retrieved October 18, 2006 from: <http://www.col.org/colweb/webdav/site/myjahiasite/shared/docs/03LMSOpenSource.pdf>
- Abel, R. (2005, May 25). *Will open source software become an important institutional strategy in higher education?* Alliance for Higher Education Competitive website. Retrieved October 31, 2007 from: http://www.a-hec.org/research/in-depth_articles/open_source0505/open_source0505_toc.html
- Anderson, T. (2004). Toward a theory of online learning. In T. Anderson & F. Elloumni (Eds.) *Theory and practice of online learning*. Athabasca, AB.: Athabasca University Press. Retrieved October 31, 2007 from: http://cde.athabascau.ca/online_book/ch2.html
- Answers.com (2005a). 'Open source movement' from Answers.com website. Retrieved November 1, 2007 from: <http://www.answers.com/open-source&r=67>
- Answers.com (2005b). 'Open source' from Answers.com website. Retrieved October 31, 2007 from: <http://www.answers.com/open-source&r=67>
- Bezroukov, N. (2005a). Open source software development as a special type of academic research (Critique of Vulgar Raymond). *First Monday*, 4(10), Retrieved November 1, 2007 from: http://firstmonday.org/issues/issue4_10/bezroukov/
- Bezroukov, N. (2005b). A second look at the cathedral and the Bazaar. *First Monday* 4(12), Retrieved November 1, 2007 from: http://firstmonday.org/issues/issue4_12/bezroukov/
- Bonk, C. J., & Cunningham, D. J. (1998). Searching for learner-centered, constructivist, and sociocultural components of collaborative educational learning tools. In Curtis J. Bonk & Kira S. King (Eds.), *Electronic Collaborators: Learner-centered technologies for literacy, apprenticeship, and discourse* (pp. 25-50). Mahwah, NJ.: Erlbaum.
- Carey, P., & Gleason, B. (2006). Vision 2010: The future of higher education business and learning applications. *Innovate*, 3(1). Retrieved October 31, 2007 from: <http://www.innovateonline.info/index.php?view=article&id=314>
- Cole, M. (1996). *Cultural Psychology: A once and future discipline*. Cambridge, MA.: Harvard University Press.
- Coppola, C., & Neelley, E. (2004). Open source – opens learning. *The R-Smart Group*. Retrieved October 31, 2007 from: <http://www.rsmart.com/assets/OpenSourceOpensLearningJuly2004.pdf>

- Downes, S. (2006a). Places to Go: Intute. *Innovate*, 3(1). Retrieved October 31, 2007 from: <http://www.innovateonline.info/index.php?view=article&id=398>
- Downes, S. (2006b). E-learning 2.0. *eLearn Magazine*. Retrieved October 31, 2007 from: <http://www.elearnmag.org/subpage.cfm?section=articles&article=29-1>
- Farrell, G. M. (2003, October). *A Virtual university for small states of the Commonwealth*. Vancouver: The Commonwealth of Learning.
- Fisher, G., Giaccardo, E., Eden, H., Sugimoto, M., & Ye, Y. (2005). Beyond binary choices: Integrating individual and social creativity. *International Journal of Human-Computer Studies*, 63(4), 482-512.
- Gleason, B. W. (2001, June 26). uPortal: A common portal reference framework. *Syllabus*, 14(12), 14-15. Retrieved October 31, 2007 from: <http://www.campus-technology.com/article.asp?id=4136>
- Godfrey, M. W., & Tu, Q. (2000). *Evolution in open source software: A case study*. Retrieved October 31, 2007 from: <http://turingmachine.org/opensource/papers/godfrey2000.pdf>
- Hartrey, P. L., & Fallon, M. A. C. (2004, November 30). IT issues & Strategic viewpoints in higher education. *Campus Technology*. Retrieved November 1, 2007 from: <http://www.campus-technology.com/article.asp?id=10308>
- Hepburn, G., & Buley, J. (2006). Getting Open Source Software into Schools: Strategies and challenges. *Innovate*, 3(1). Retrieved October 31, 2007 from: <http://www.innovateonline.info/index.php?view=article&id=323>
- Hilton, J. L. (2005). In praise of sharing. *EDUCAUSE Review*, 40(3), 72-73.
- Huang, K., Dong, Y., & Ge, X. (2006). From, by, and for the OSSD: Software engineering education using an open source software approach and ideal. *Innovate*, 3(1). Retrieved October 31, 2007 from: <http://www.innovateonline.info/index.php?view=article&id=32>
- Iiyoshi, T., Richardson, C., & McGrath, O. (2006). Harnessing open technologies to promote open educational knowledge sharing. *Innovate*, 3(1). Retrieved October 31, 2007 from: <http://www.innovateonline.info/index.php?view=article&id=339&highlight=Iiyoshi>
- Internet Mail Standard (2006). *RFCs and Internet Drafts*. Internet Mail Consortium webpage. Retrieved October 31, 2007 from: <http://www.imc.org/mail-standards.html>
- Johnson, K. (2005). *A descriptive process model for open-source software development*. Retrieved December 16, 2005, from <http://sern.ucalgary.ca/students/theses/KimJohnson/chapter2.htm>
- Kapor, M. (2005). How is open source special? *EDUCAUSE Review*, 40(2), 72-73.

- Kivekäs, O. (2005). The Bazaar methodology of software development. *Proceedings at Research Seminar on Open Source and Free Software*, University of Helsinki, Department of Computer Science, May 15, Helsinki, Finland.
- Linux Online (2005). *What is Linux?* Retrieved November 1, 2007 from:
<http://www.linux.org/info/index.html>
- Machado, C., & Thompson, K. (2005). The adoption of open sources within higher education in Europe and a dissemination case study. *Turkish Online Journal of Distance Education*, 6(1). Retrieved November 1, 2007 from:
<http://tojde.anadolu.edu.tr/tojde17/articles/carlos.htm>
- Mahoney, M. J. (2004). Constructivism. Retrieved February 12, 2006, from
<http://www.constructivism123.com>
- Moodle (2005a). *Moodle website*. Retrieved November 1, 2007 from: <http://moodle.org/>
- Moodle (2005b). *Moodle UTF-8 language packs*. Retrieved November 1, 2007 from:
<http://download.moodle.org/lang/?MoodleSession=8b50ac297a877da6658fb575189e95f2>
- Moodle (2006). *Moodle community*. Retrieved November 1, 2007 from: <http://moodle.org/>
- Morin, R. (1998, January). Of cathedral and Bazaar. *SunExpert Magazine*, 9(1). Retrieved November 1, 2007 from: <http://www.macperl.com/depts/articles/OCaB.html>
- Open Source Initiative (2006). Introduction. *Open Source Initiative website*. Retrieved December 28, 2006 from: <http://opensource.org/node/104>
- Pfaffenberger, B. (2000, March 2). *Linux in Higher Education: Open source, open minds, social justice*. Retrieved November 1, 2007 from: <http://www.linuxjournal.com/article/5071>
- Quamen, H. (2006). Looking Toward the Future: A case study of open source software in the humanities. *Innovate*, 3(1). Retrieved October 31, 2007 from:
<http://www.innovateonline.info/index.php?view=article&id=325>
- Raymond, E. (1997). *The cathedral and the Bazaar*. Retrieved December 9, 2005 from
<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>
- RFC.net (2006). *What is an RFC? RFC.Net website*. Retrieved October 31, 2007 from:
<http://www.rfc.net/>
- Rhodes, F. (1999). *The Gilon declaration*. Retrieved December 26, 2005, from
<http://www.glion.org/?a=6202&p=1512>
- Robles, G. (2004). *A software engineering approach to Libre software*. Germany: Open Source Jahrbuch. Retrieved December 16, 2005, from
<http://www.opensourcejahrbuch.de/2004/pdfs/III-3-Robles.pdf>

- Room 17. (2005). *Bazaar source*. Retrieved November 1, 2007 from:
http://www.room17.com/ramblings/bazaar_src.shtml
- Rourke, L., & Anderson, T. (2002). Using Web-based, group communication systems to support case study learning at a distance. *International Review of Research Open and Distance Learning*, 3(2), Retrieved November 1, 2007 from:
<http://www.irrodl.org/index.php/irrodl/article/view/107/186>
- Sakai (2005). *About Sakai*. Retrieved October 31, 2007 from:
http://www.sakaiproject.org/index.php?option=com_content&task=view&id=103&Itemid=208
- Sakai (2006). *The Sakai Partners Program* . Retrieved October 17, 2006, from
http://sakaiproject.org/index.php?option=com_content&task=view&id=233&Itemid=462
- Stephenson, R. (2006). Open source/Open course learning: Lessons for educators from free and open source software. *Innovate*, 3(1). Retrieved October 17, 2006, from
<http://www.innovateonline.info/index.php?view=article&id=345>
- uPortal (2006). *uPortal in Production*. Retrieved October 31, 2007 from: <http://uportal.org/who-prod.html>
- Weber, S. (2004). *The success of open source*. Cambridge, MA.: Harvard University Press.
- Wheeler, B. (2004). Open source 2007: How did this happen? *EDUCAUSE Review*, 39(4), 12-27.
- Wheeler, D. (2006, April 16). *Why open source software / free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers!* Retrieved November 1, 2007 from:
http://www.dwheeler.com/oss_fs_why.html
- Wikipedia (2005). 'Wikipedia' *wiki entry*. Retrieved November 1, 2007 from:
<http://en.wikipedia.org/wiki/Wikipedia>
- Wiley, D. (2006). Open source, openness, and higher education. *Innovate*, 3(1). Retrieved November 1, 2007 from: <http://www.innovateonline.info/index.php?view=article&id=354>
- Yanosky, R., Harris, M., & Zastrocky, M. (2003). *Higher-education E-learning meets open source*. Gartner.com website. Retrieved December 28, 2005 from:
<http://www4.gartner.com/LogNReg/manualLogin>

