Open courseware and computer science education

Article *in* Journal of Computing Sciences in Colleges · January 2004

4 authors:

Kwok-Bun Yue
University of Houston - Clear Lake
42 PUBLICATIONS **447** CITATIONS

T. Andrew Yang
University of Houston - Clear Lake
62 PUBLICATIONS **1,123** CITATIONS

Wei Ding
University of Massachusetts Boston
112 PUBLICATIONS **5,521** CITATIONS

Ping Chen
Beijing Forestry University
286 PUBLICATIONS **8,799** CITATIONS

OPEN COURSEWARE AND COMPUTER SCIENCE EDUCATION

Kwok-Bun Yue, T. Andrew Yang, Wei Ding
University of Houston-Clear Lake
2700 Bay Area Boulevard, Houston, TX 77058, USA
{yue|yang|ding}@cl.uh.edu

Ping Chen
University of Houston-Downtown
One Main Street, Houston, TX 77002, USA
chenp@uhd.edu

ABSTRACT

The recent enthusiastic reception of the MIT *OpenCourseWare* (*OCW*) project has significantly improved the general awareness of Open Courseware (OC). However, many other lesser known projects and resources can also be classified as OC. The OC movement can potentially provide a vast pool of resources to satisfy diverse needs of Computer Science (CS) educators. However, there are only limited discussions on the possible meanings of OC to CS education. This paper elaborates several important facets of OC. It describes how CS educators can utilize raw educational materials from OC and how OC can support a continuum of approaches on constructing courseware. The impact of OC on CS educators will likely be greater than that of Open Source Software (OSS), since CS educators are more likely *developers* of course contents but only *users* of OSS. Thus, this paper suggests deeper and broader studies on the opportunities and challenges of OC provided to CS education.

1. INTRODUCTION

The MIT *OpenCourseWare* (*OCW*) project was launched in 2001 to eventually 'open source' *all* 2000 MIT courses by 2008 [14]. That a top-notch university would 'give away' what some considered the most important intellectual property of a university has attracted a lot of attention to the concept of open courseware (OC). (As an example of OC, please see Appendix A for more details on *MIT OCW*.) Since then, most educators and learners have enthusiastically applauded *OCW*. For examples, Baldi studied the *OCW* model from various angles [2] and Newmarch proposed alternatives to courseware's intellectual property based on the *secrecy* model for university education [17]. Newmarch argued for the OC model by elaborating that courses are not just courseware, but also include personal contact, timetable, guidance, feedback, assessment and certification [17].

Instructors have long put their courseware up in the Web for easy and free access. The recent success of Open Source Software (OSS), such as Linux and Apache, has further spurred many projects to adapt the OSS approach for open contents and open courseware. MIT *OCW* is just one of the most visible OC projects. These OC projects and approaches can provide a vast pool of resources to satisfy diverse needs of Computer Science (CS) educators. However, there are only limited discussions on the possible meanings of OC to CS education. This paper discusses how different facets of OC may assist CS educators in courseware development. It describes how CS educators can utilize raw educational materials from OC and how OC can support a continuum of approaches on constructing courseware.

In Section 2, we discuss the various facets and interpretations of OC, thus providing a more holistic view. Section 3 describes how OC can be used as raw educational materials and for constructing courseware. We draw our conclusions in Section 4.

2. DIFFERENT FACETS OF COURSEWARE

OC loosely refers to the open and free publications of course materials, accessible usually through the Web. However, different people may be referring to different concepts when using the term "open courseware." In this section, we first examine the licensing issues, and then discuss OC's different facets: as a course development model, as courseware development software, and as learning objects and standards. Together, they provide a more complete perspective on the nature and complexity of OC.

2.1 Copyright Licenses

OC usually has open copyright licenses similar to that of OSS, in order to enable the public to access, copy and distribute the content materials. Educators can usually find relevant resources in the Web that they may want to incorporate into their courses. However, copyright consideration is a serious roadblock even if the original intention of the content author was meant to provide the resource for free access and use. This is especially the case since creative works are *automatically* copyrighted in the United States and many other countries [8]. A set of well-defined open copyright licenses, such as those provided by Creative Commons [7], allows the users to confidently use the resources in the designated manner. For example, MIT *OCW* uses four of the eleven available Creative Common licenses: attribution, share alike, non-commercial and exceptions [9, 14].

2.2 OC as Courseware Development Model

A central theme of OSS is a software *development* model based on a Web community to allow programmers to rapidly adapt, improve and fix bugs for the software [12, 15, 22]. The basic premise of OSS is 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" [20]. When people refer to OC, they may be describing a similar community-based courseware development model. A well-known OC project that uses an OSS community-based development approach is Rice University's Connexions [6]. In Connexions, content developers can create projects to collaborate on developing *modules*, which can then be selected to assemble courses. Students and other community users may then access the contents to provide feedback for their improvement. However, only few OC projects use such a community-based approach for content development. For example, MIT *OCW* contents are basically developed by MIT faculty members.

As CS educators, one consideration is whether to adopt an OC approach for developing courseware. Several experiments have been reported, such as [1] and [26]. There are many different approaches in adopting OC in courseware development. On one end, an individual instructor may build a small Web community on developing selected courseware materials with minimal use of collaborative software and OC standards. On the other end, it can be several universities collaborating on developing full courseware on a series of courses with extensive use of collaborative software and OC standards. Although this is a very important issue, the details of the different approaches and their relative merits will not be considered here because of space.

2.3 OC as Courseware Development Software

OC may also be referring to the use of OSS usually known as Course Management System (CMS) or Learning Management System (LMS), to design, develop and deliver courseware [11]. For example, *Moodle* is an open source CMS based on the social constructionist pedagogy, which is currently used by the United Nations Educational, Scientific and Cultural Organization (UNESCO) in its *Open Educational Resources* (OER) initiative [16]. These open source courseware development software can be used as alternatives to their better-known commercial siblings, such as WebCT and Blackboard. For a good glimpse on open source CMS/LMS, the commonwealth of learning has identified 35 such products and evaluated five of them [4].

2.4 OC as Learning Objects and Standards

This aspect of OC refers to courses designed as inter-related learning objects likely to be defined by open standards. Important example standards are *Open Knowledge Initiative* (OKI) [19] and *Sharable Content Object Reference Model* (SCORM) [23]. These standards aim to improve the reusability and interoperability of OC. For example, OKI is “an open and extensible architecture that specifies how the components of an educational software environment communicate with each other and with other enterprise systems” [19]. SCORM defines a Web-based learning content aggregation model and run-time environment for learning objects to improve compatibility.

Although standards increase the reusability and interoperability of the courseware, their adherence usually means more work and less flexibility, and may thus not be justified in a given situation. For examples, they may be very suitable for highly collaborative projects with participants from many organizations, but not for a rapidly changing local course taught by a single instructor.

3. COURSEWARE DEVELOPMENT

To highlight the impact of OC on CS educators, a highly simplified view on the two major activities of constructing courseware is shown in Figure 1.

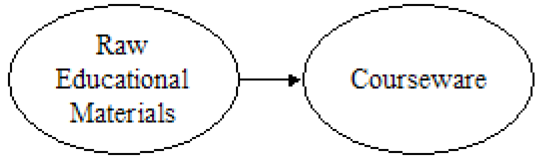

Figure 1. A Simplified View of Courseware Development

Instructors first collect and prepare raw educational materials, such as bits and pieces of examples, exercises, resource links, case studies, lecture notes, etc. These raw materials are then synthesized into integrated courseware to support the requirements and goals of the courses. The instructor may also select a mechanism to host the courseware for effective dissemination, communication and management. We will briefly discuss how OC may contribute to these two steps.

3.1 Raw Educational Materials

Using a popular search engine can produce numerous raw contents that are relevant to a particular course. However, several issues restrict its effective uses.

(a) The resources may have unclear or restrictive copyrights.

(b) They may not be designed for instructional purpose and may have uneven quality.

(c) They may also be too scattered for effective use and integration.

Existing OC projects solve many, but not necessarily all, of these issues.

There exist several types of OC projects, which are potential sources of raw educational materials. Among the most visible are *repositories* of OC courses by universities and other institutes, such as MIT *OCW* and Rice's *Connexions*, usually with support commitment from the top-level administrations of the organizations. These sites usually have OC copyright licenses and contain a good collection of courses. They also have good built-in quality control mechanism. For example, MIT *OCW* is developed by its reputed faculty, and *Connexions* has a community based post-publication peer review mechanism.

Another type of OC projects consists of the many scattered course Web sites published directly by individual instructors. These sites tend to exhibit varying quality, comprehension and richness. For example, Michael Rappa maintains a popular and rich open courseware on managing the digital enterprise [21]. Searching for appropriate courseware for a given course can be time consuming, but the effort is worthwhile if a suitable match is found.

The last type of OC projects contains Web sites that provide freely accessible repositories of annotations and reviews on external resources. A popular example is *Merlot* [13], which is an open site with annotated links to online learning materials, designed primarily for faculty and students of higher education. For CS educators, the Computer Science Teaching Center (CSTC) [5], the Science, Mathematics, Engineering and Technology Education (SMETE) Digital Library [25], and the Computing and Information Technology Interactive Digital Educational Library (Citidel) [3] are among the most relevant. However, instructors may want to be extra cautious in copyright consideration when using the external resources linked to by these repositories. Although the annotations and reviews to these external resources usually have an OC copyright license, the *actual* external resources may *not*.

3.2 Approaches of Courseware Construction

There is a continuum of approaches instructors may use OC to develop and host their courseware. They can select an approach in this continuum that best satisfies the requirements, goals, audiences and existing resources of the courses.

In one end, instructors may simply *refer* to an OC site where the complete courseware is available (the *direct-linking* approach). Supplementary materials, such as assignments, examinations, discussion groups, etc, may then be added locally. In fact, the *Wired* magazine has reported efforts outside the United States on using only MIT *OCW* as the main content for supporting local courses [10]. As reported, some of these efforts may not even be legal because of their commercial nature.

The obvious advantages of this simple approach are the built-in quality of the OC site and the relatively minimal efforts required of the instructor. The primary weakness of this approach is that instructors lose some or all control over the contents and their availability. Note that there are differences in terms of flexibility of course constructions among different OC sites. MIT *OCW*, for example, does not allow external users, instructors or students, to modify their online courseware hosted in *OCW*. On the other hand, Rice's *Connexions* allow instructors to construct their own courses by assembling existing modules. If instructors select to develop the modules themselves, they will have even more control over course construction. Even so, fine-grain customizations, such as horizontal versioning, which may be needed to satisfy different local requirements and goals of different courses, are not available.

In the other end of the continuum, instructors may copy, modify and paste raw educational materials obtained from OC sites to construct their courseware locally. External OC materials are thus embedded into the local courseware. This *copy-and-paste* approach usually means more work for the instructors but provides the highest degree of flexibility and customization. However, two important issues should be considered with this approach:

(a) Licensing issues: Although OC licenses are similar, there are important differences and instructors will need to pay attention to the details. For example, many OC licenses, including those of MIT *OCW*, include the *attribution* term. The attribution license by Creative Commons allows users to "copy, distribute, display, and perform your copyrighted work — and derivative works based upon it — but only if they give you credit" [9]. That means the instructors will need to include a way to pay attribution to the original OC work if they are embedded into the local courseware. As another example, the infrequently used *No Derivative Works* license of Creative commons allows others to "copy, distribute, display, and perform only verbatim copies of your work, not derivative works based upon it" [9]. This may mean that the instructors cannot embed the original OC materials within their local courseware as they can be considered as derivatives.

(b) Content currency issues: The second issue is the currency of the educational materials. This is the familiar redundancy problem in computer science, as the original OC materials are now embedded as a possibly modified copy in the local courseware.

When the original material is updated, its embedded copy in the courseware may not be updated accordingly and thus become a stale copy. This is also similar to the need of upgrading a derivative of an older version of software to base on a newer version. Unfortunately, there is no easy answer to this question. If the original work is not modified and is openly published using open standards, proper CMS with active linking and embedding capability can be used to drastically simplify the update task. That is not necessarily the case for most available OC. A mix of manual and semi-automatic method can be devised. For example, an instructor may use software to keep track of all the sources of the original educational materials. Regular automatic checking can be scheduled. All stale resources can then be reported back to the instructors for manual update.

Note that the *direct-linking* approach has a different kind of currency issue since the external OC sites may not be updated frequently enough to keep abreast of the advance of the course subject.

In between *direct-linking* and *copy-and-paste*, there exist other approaches in the continuum. For example, an instructor may directly link only to case studies of external OC sites, but develop lecture notes locally by the *copy-and-paste* approach. No matter which approaches are selected, we expect the instructors must address both the licensing and currency issues.

4. CONCLUSIONS

In this paper, we have discussed the various facets of OC and how CS educators may benefit from it. A continuum of approaches and related issues on using OC has also been discussed.

Despite the existence of an already sizable amount of resources and projects, it is only the very early phase of the OC movement. There are numerous ongoing projects that promise to significantly extend the depth and scope of OC in every facet. For an example, the *Sakai* project pulls together four major universities, the *uPortal* Consortium, and the *Open Knowledge Initiative* (OKI) to "integrate and synchronize their considerable educational software into a modular, pre-integrated collection of open source tools for courseware development" [24]. In fact, our own team is also working on building a prototype of an open content community for the collaborative development of OC educational materials in a way similar to sourceforge.net for OSS [27].

The success of OSS has prompted CS educators to study its impact on CS educators [18]. In fact, the use of OSS in CS curriculum is a constant theme in recent conferences of the Consortium of Computing Sciences in Colleges. Despite this attention, CS educators are basically *users* of OSS. Very few of them are *developers* of OSS. On the other hand, a significant part of the job of CS educators is to *develop* course contents. OC impacts CS educators both as users and developers. Its significance can be significantly greater than that of OSS.

Thus, this paper suggests deeper and broader studies on the opportunities and challenges of OC to CS education. For example, the issue of the relative merits of adopting an OC model for courseware development is highly relevant to CS educators. This will become even more so in the near future as OC becomes more ubiquitous. In fact, our team has embarked on another project on developing Internet application development courseware using an OC approach and will report our findings in another article.

ACKNOWLEDGEMENT

This work is partially supported by the faculty development leave fund of the University of Houston-Clear Lake, the Institute of Space Systems Operations and the National Science Foundation (Grant DUE-0311592).

REFERENCES

[1] Ala-Mutka, K. & Mikkonen, T., Experiences with Distributed Open Source Courses, *Proceedings of the 8th International Conference of European University Information Systems*, Portugal, 26-37, June 19-22, 2002.

[2] Baldi, S., et. al., Open Courseware vs. Open Source Software - a Critical Comparison. *10th European Conference on Information Systems*, Gdan'sk, Poland, 1375-1383, 2002.

[3] Computing and Information Technology Interactive Digital Educational Library's home page, 2004, http://www.citidel.org/, retrieved June 28, 2004.

[4] The Commonwealth of Learning, *COL LMS Open Source*, 2004, http://www.col.org/Consultancies/03LMSOpenSource.pdf, retrieved June 28, 2004.

[5] Computer Science Teaching Center's home page, 2004, http://www.cstc.org/, retrieved June 28, 2004.

[6] Connexions home page, 2004, http://cnx.rice.edu/, retrieved June 28, 2004.

[7] Creative Commons home page, 2004, http://creativecommons.org/, retrieved June 28, 2004.

[8] Creative Commons' frequently asked questions, 2004, http://creativecommons.org/faq, retrieved June 28, 2004.

[9] Creative Commons, *License Explained*, 2004, http://creativecommons.org/learn/licenses/, retrieved June 28, 2004.

[10] Diamond, D., MIT Everyware, *Wired*, 11, (9) http://www.wired.com/wired/archive/11.09/mit.html, retrieved June 28, 2004.

[11] Gardler H., Gardler, R. & Singh.L, The Use of Open Source Information and Communications Technology (ICT) in Education, *Free, Libré and Open Source" (FLOS) Software Conference*, 2003, http://cis.saafe.org/publications/papers/flosCaribbean2003.html, retrieved June 28, 2004.

[12] Kim, E., *An introduction to open source communities*, April 2003, http://opensource.mit.edu/papers/blueoxen.pdf, retrieved June 28, 2004.

[13] Multimedia Educational Resources Learning and Online Teaching's (Merlot) home Page, 2004, http://www.merlot.org/Home.po, retrieved June 28, 2004.

[14] MIT OpenCourseWare's home page, 2004, http://ocw.mit.edu/index.html, retrieved June 28, 2004.

[15] Mockus, A., Fielding, R. & Herbsleb J., Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Software Engineering Methodology* 11, (3), 309-346, 2002.

[16] Moodle's home page, 2004, http://moodle.org/, retrieved June 28, 2004.

[17] Newmarch J., Lesson from Open Source: Intellectual property and courseware, *First Monday*, 6, (6), June 2001, http://firstmonday.org/issues/issue6_6/newmarch/index.html, retrieved June 28, 2004.
[18] O'Hara, K. & Kay, J., Open source software and computer science education, *the Journal of Computing in Small Colleges*, 18, (3), 1-7, February 2003.
[19] Open Knowledge Initiative's home page, 2004, http://web.mit.edu/oki/, retrieved June 28, 2004.
[20] Open Source Initiative's home page, 2004, http://www.opensource.org/index.php, retrieved June 28, 2004.
[21] Michael Rappa, *Managing the digital enterprise*, 2004, http://digitalenterprise.org/index.html, retrieved June 28, 2004.
[22] Raymond, E., *The cathedral and the bazaar*, version 3.0., 2000, http://www.catb.org/%7Eesr/writings/cathedral-bazaar/cathedral-bazaar/, retrieved June 28, 2004.
[23] Sharable Content Object Reference Model Initiative's home page, 2004, http://www.adlnet.org/index.cfm?fuseaction=scormabt, retrieved June 28, 2004.
[24] The Sakai project's home page, 2004, http://www.sakaiproject.org/, retrieved June 28, 2004.
[25] SMETE Digital Library's home page, 2004, http://www.smete.org/, retrieved June 28, 2004.
[26] Smith, J., Moore, G., Wingenbach, G. & Straquadine G., An "Open Courseware" Approach to Teaching an Online Course in Agriculture, poster session presented at the *29th National Agricultural Education Research Conference*, Las Vegas, 2002, http://www.depts.ttu.edu/aged/research/smithopen.pdf, retrieved June 28, 2004.
[27] Yue, K., Yang, T., Ding, W. & Chen, P., A model for open content communities to support effective learning and teaching, *Proceedings of the IADIS International Conference on Web-based Communities*, 533-536, Lisbon, Portugal, April 2004.

APPENDIX A THE MIT'S OPENCOURSEWARE PROJECT

The MIT's OpenCourseWare (OCW) Project [14] is one of the most influential open courseware (OC) projects. It is also a good example for illustrating the nature and variety of OC. MIT OCW is a "a free and open educational resource for faculty, students, and self-learners around the world." It is a publication of MIT course materials and does not require any registration for access.

Currently, OCW hosts 701 MIT courses representing 33 academic disciplines and all five of MIT's schools. The plan is to host all 2000 MIT courses by 2008. There are currently 88 courses in the area of computer engineering and computer science. These courses cover a lot of areas in both undergraduate and graduate levels. The contents of courses vary and some courses have much more resource than others. For example, "6.170 Laboratory in Software Engineering, Fall 2001" is one of the most popular courses at MIT *OCW*. It includes lecture notes, reading, tools, assignments, projects, etc. In fact, the project on "Gizmoball" has become a very popular design project, frequently discussed by CS educators.

Instructors and students can freely access MIT *OCW* courses, which are copyrighted through open source licenses of the Creative Commons. However, *OCW* is not a degree-granting or certificate-granting activity and thus does not provide access to MIT faculty.