

Open Source Course Management Systems: a Case Study

Eric Rémy

Randolph-Macon Woman's College

2500 Rivermont Ave

Lynchburg, VA, 24503

(434)-947-8000

eremy@rmwc.edu

Abstract

In Fall 2003, Randolph-Macon Woman's College rolled out Claroline, an Open Source course management system for all the classes on campus. This document will cover some background on both Open Source in general and course management systems in specific, discuss technical challenges in the introduction and integration of the system and give some survey and usage results over the past two years at R-MWC.

Background:

Randolph-Macon Woman's College is a four-year traditional liberal arts college for women in Lynchburg, Virginia. Founded in 1893, the college now enrolls about 760 students from 39 different countries, of which about 700 are traditional students and 60 are "Prime Time" adult learners. The college is residential and has no distance education department. A student-faculty ratio of 9:1 results in very small classes and a great deal of direct student-faculty interaction.

The college did not have a course management system (CMS) until two years ago. There was a perception by the faculty that "Blackboard was for big schools" and wasn't necessary for a college where students live on or near campus and everyone knows everyone else by their first name. On-campus demos of Blackboard and WebCT in 2002 drew some limited interest by a group of faculty, but a budget crunch during that year coupled with price hikes by the two companies left some faculty questioning the wisdom of locking R-MWC into paying a lot of money for a piece of software few people would use.

An additional complication was our use of SCT PowerCampus, the college's student information system (SIS). Neither Blackboard nor WebCT salespeople seemed knowledgeable about interfacing their products with PowerCampus, and a request for a list of schools running the PowerCampus and Blackboard/WebCT combination drew no responses from either company.

PowerCampus ships with a web interface called IQWeb, which introduced some CMS functionality in 2002 (file sharing, student mailing/announcements, gradebook). After some testing, it was determined that this package was not stable and capable enough to use for production.

Open Source Software

One of the biggest stories in computer technology over the past 5 years has been the phenomenal growth of Open Source software projects. The best known of these have been the Linux operating system and the Apache web server which have grown into serious competitors for Microsoft,

Sun and other commercial companies, but common Open Source software also includes office suites (OpenOffice), web browsers (Mozilla/Firefox), image editing programs (The GIMP) and many others.

Open Source software differs from normal commercial software only in that the human readable source code is available to the user so that they can inspect and modify it. Many Open Source projects also have computer-readable executable files available for easy installation. In contrast, most commercial companies will carefully guard the source code for their products, treating the source as a trade secret. The user receives only a computer-readable executable and may not look at or modify the internal code.

There are a number of different Open Source licenses (see Appendix 1) with some important distinctions in how the code may be used and modified. Most but not all of these packages cost nothing. (Free Software is a confusing term here-see the Appendix)

The development philosophy of Open Source is different than for closed source products. Rather than having a dedicated team of developers working at a central company, carefully managed to produce a saleable product, the Open Source ideal is that thousands of users will download and inspect the code as it is released. The users themselves will find the bugs and fix them, as well as adding features that they feel are necessary. These small fixes should occur quickly, enabling rapid and steady growth in a software package as a whole: "release early, release often". The Internet allows users distributed around the globe to collaborate almost as easily as if they were in the same room. The best known explanation of this philosophy is Eric Raymond's *The Cathedral and the Bazaar*.

Many companies are beginning to open some of their source code as part of their overall strategy. Blackboard has opened parts of their closed source CMS so that users can connect other pieces of software such as a virtual laboratory.

Open Source Course Management Systems

Over the past few years, a number of schools have written their own course management systems. Frequently, these have been treated as commercial entities, with the college spinning off the software and selling it to other colleges and universities. The best known example is Prometheus, which was developed at George Washington University. After developing the software in-house, GWU sold the software to 65 schools before Blackboard bought the project in 2002. Prometheus was itself somewhat Open Source, allowing paying customers to view and modify the source code under a model they referred to as "Community Source".

Other schools have chosen instead to develop similar products under Open Source licenses and give the code away. The best known of these is Sakai, a collaboration between Indiana University, the University of Michigan, Stanford, and MIT, supported by a large grant by the Mellon Foundation.

Open Source does not prevent these projects from going commercial. Both Dokeos and Moodle are developed by companies which sell support, customization and site hosting even though they allow the full source code to be downloaded free of charge.

Advantages to Open Source

1. Users can modify the code to support what they need.
2. Turnaround times on modifications can be much faster: no waiting for Blackboard to add a desired feature.
3. A potentially much larger development community: chances are that someone somewhere has the same problem and may have already fixed it.
4. Potentially better support: even if the support avenue is an online forum, you generally talk directly to the primary developers and not a poorly trained support technician.
5. Frequently (but not always) zero cost.

Disadvantages to Open Source

1. Features are not at the level of Blackboard/WebCT. Dokeos, for example, does not have a virtual whiteboard that may be shared by many users, and the audio/video conferencing tool is nowhere near the capability of the ones in commercial products.
2. Users can modify the software. This often means they **must** modify the software: chances are that it won't do what they want out of the box. Integration with other software products is almost nonexistent for most Open Source programs.
3. Users will require a higher level of internal support. Just running Dokeos could be done by any competent IT person, but getting around the quirks may require a trip into the database.
4. Supervisors will have a smaller pool of potential experienced workers. Placing an advertisement for a Blackboard support specialist will probably bring a number of responses. Placing one for a Sakai specialist won't, at least at this stage.
5. Documentation may be poor to nonexistent.
6. Users may have no one to blame if something goes wrong.

Implementation of Claroline at R-MWC

We began consideration of alternatives to Blackboard and WebCT after the rather underwhelming interest by the faculty left us pondering the wisdom of investing in an expensive product. I stumbled across Claroline, Moodle and .LRN while looking for other information and was interested enough to begin evaluation. (Sakai was not available at the time of our evaluation). All of these applications are in use at many large colleges and universities; a basic technical comparison between the three along with Dokeos and Sakai is shown below.

Name	Home institution	Language	Server	Database	License
Claroline	Universite Catholique de Louvain	PHP	Apache	MySQL	Gnu Public License (GPL)
Dokeos	Commercial	PHP	Apache	MySQL	GPL
.LRN	MIT Sloan	Tcl	AOL Server	Oracle or PostgreSQL	GPL
Moodle	Curtin University of Technology	PHP	Apache	MySQL or PostgreSQL	GPL
Sakai	Multiple	Java	Tomcat	HSQldb, MySQL or Oracle	Educational Community License

In our evaluation, we focused both on possible usage as well as ease of use and support. R-MWC has a single person in charge of instructional technology who has multiple responsibilities and if we would be using a product with limited support simplicity was a major concern. Since R-MWC has no distance education courses and very small classes, a sophisticated conferencing system, advanced chat features or powerful quiz modules were not priorities. We felt that usage would primarily be document sharing, discussion forums, class announcements as well as mass email, web link management and class scheduling. Multiple language support was a major concern as R-MWC has a large international population and offers courses in 9 languages. .LRN was eliminated early on due to its additional complexity over Claroline or Moodle. Moodle and Claroline were roughly comparable at the time of evaluation: both supported the features above as well as having full translations for all the languages of interest. Claroline was eventually chosen more or less by default: it was the first test system installed and I was somewhat more familiar with it. I suspect Moodle would have been equally as effective.

Claroline presents a very simple user interface for both faculty and students. (Please note screenshots are of our current Dokeos 1.5.4 installation, not the older Claroline). On login users see a list of all the classes in which they are enrolled. Selecting a course shows a list of the available tools for that class. Faculty may activate or hide any tools they wish. Tools available in the version R-MWC is running currently (Dokeos 1.5.4) include

- Agenda: calendar items
- Announcements: class email and posting
- Chat: web based
- Conference: marries chat, audio/video streaming and document display
- Course description: online syllabus builder



- Documents
- Discussion forums
- Dropbox: faculty and student document transfer
- Groups
- Learning Path: SCORM module import, view and build
- Links
- Tests
- User list
- User tracking

Claroline will work on any system that can run the Apache web server, MySQL database, and PHP scripting language. While some sites run it under Windows, we chose to follow the majority and run it under Unix, in our case RedHat 9.0. The server is a Penguin Computing dual Xeon machine with 1GB of RAM and 300GB disk space. This is overkill for Claroline at R-MWC; a similar size machine serves much larger schools. (Claroline scales fairly well: UCL supports more than 20,000 users on their system.) Claroline will work on virtually any browser as it makes virtually no use of Java, JavaScript, plugins or other technologies.

Two major obstacles had to be overcome before rolling out Claroline to the community as a whole

1. Claroline works on a "build it and they will come" model where faculty are responsible for creating their classes and students are responsible for enrolling in those classes. Given the relatively low interest level among the faculty, this model was not going to work at R-MWC; we needed to create all users and classes ahead of time and keep enrollment updated.
2. LDAP support was poor in Claroline 1.4, meaning that users would need to have a separate password for the system.

The open nature of Claroline made both obstacles relatively easy to overcome. For auto-enrollment, our DB admin wrote a series of views into the PowerCampus database listing current faculty, students, courses, and course enrollments and presented them as simple web pages. For user and course creation, I wrote a series of PHP script wrappers around existing Claroline code which loaded these web pages and then used the existing code to insert the items. Enrollment was handled by a custom PHP script that simply wrote the correct values for the user-course pairs directly into the Claroline database. This took less than a week of our time, even while working on other projects.

The screenshot shows a web interface for Randolph-Macon Woman's College. At the top, there's a header with the college name and user navigation links: "Eric Remy : My courses list | My profile | My agenda | Logout". Below that, the course title is "INTERDISCIPLINARY FORUM (Fall 2004) IST103A - Hullum, Jan". A breadcrumb trail reads "Randolph-Macon Woman's College > IST103A".

The main content area is divided into several sections:

- Help**: A small icon and text.
- Add introduction text**: A link to add introductory text.
- Course description**: A link with a "Hide" option below it.
- Documents**: A link with a "Hide" option below it.
- Announcements**: A link with a "Hide" option below it.
- Dropbox**: A link with a "Hide" option below it.
- Groups**: A link with a "Hide" option below it.
- Agenda**: A link with a "Hide" option below it.
- Links**: A link with a "Hide" option below it.
- Forums**: A link with a "Hide" option below it.
- Users**: A link with a "Hide" option below it.
- Course E**: A link with a "Hide" option below it.

Below these is a section for **Teachers only**, containing:

- Tracking**: A link with a "Remove" option below it.
- Course settings**: A link with a "Remove" option below it.
- Add a link**: A link with a "Remove" option below it.
- Import old**: A link with a "Remove" option below it.

At the bottom, there's a section for **Hidden tools and links**:

- Learning Path**: A link with "Show" and "Remove" options below it.
- Tests**: A link with a "Show" and "Remove" option below it.

A fix for the password issue was also fairly simple. We rolled out a new campus portal (Academy uPortal from Unicon) over the same summer we launched Claroline. A primary goal of the portal was a single sign on for all major systems at R-MWC such as webmail and IQWeb, so obviously course management needed to be included as well. Our portal administrator created a channel for Claroline and had the portal pass the username and a token to Claroline when a user who had successfully authenticated against the portal requested the Claroline channel. The token is a hash based on the username as well as some secret information. I then modified the login code for Claroline to check for a valid token and bypass the login screen if the token passed.

The end result is a system that is virtually transparent to the students and faculty. When they go to the portal, clicking on the myCourses tab will immediately present them with a list of the courses in which they are currently enrolled. The system originally ran the enrollment script every morning at 3:00AM so the list is constantly updated for add/drops, although the system is fast enough that we have since gone to hourly updates- matching 750 users and 450 courses into roughly 4500 enrollments takes less than a minute. The enrollment script originally did not permit students to self-enroll into a class, but this has since been changed due to user feedback.

We made some additional changes as well, such as including pictures of all students in the user list. One notable example shows the Open Source development philosophy of user contributed code: Claroline lacked a single master list of all class agenda items for a given user, so students were stuck going to each class to see events in the agenda tool. I wrote a simple calendar function which displayed all of these items on a single master calendar with links directly to the item. I posted a comment in the Claroline forum asking if anyone else wanted the code. A number of schools as well as one of the primary developers did. I sent to code to anyone who asked, and a modified version was built into the next formal release.

GENERAL CHEMISTRY LABORATORY
CHEM105LC - Bare, William

<< October 2003 >>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1 CHEM105LC: Second Lab Quiz CHEM105LC: Zinc Chloride Lab CHEM105LC: Copper Salt Lab Due	2 SDAN187A: Class project brainstorming session	3 EVST101A: Concept Paper Outline EVST101A: Bladewater Creek Stream Project assigned CHEM105C: First Exam	4
5	6 EVST101A: Student review and study session for Exam #1	7 EVST101A: Review Session for Exam #1 SDAN187A: Reaction paper	8 EVST101A: Exam #1	9	10 EVST101A: Bladewater Creek Stream Groups announced in	11

Claroline was introduced to the faculty in a series of voluntary workshops before the start of the Fall 2003 semester. As the documentation from the Claroline site was for an older version of the software without some of our enhancements, I wrote a 20 page manual for the program with step-by-step instructions and screenshots and distributed paper copies to all faculty members. A shorter student manual was made available online. It took roughly 90 minutes to introduce faculty to the features of the software leaving 30 minutes or so for them to practice. Roughly half of the faculty attended one of the workshops. Faculty usage was encouraged by the college requiring faculty to post their syllabi for all of their classes online.

Overall, the introduction went fairly smoothly, but some issues cropped up early. Notable ones included:

- Claroline was limited by default to 2MB document uploads due to a setting in the PHP scripting language, far too small for faculty who quickly decided they wanted to send 100MB PowerPoint slide shows to their students.

- Something about Claroline would occasionally cause Apache to hang after a long period of use. I never located the source of the problem, but restarting Apache every morning at 2AM cured the problem. (This bug has not reoccurred since the Dokeos upgrade, coupled with an OS upgrade from RedHat 9 to Fedora Core 2).
- A bug in the way Claroline set a login cookie would have users appear to log in, but get kicked into guest mode after the login page. While isolating the bug took a morning of frustration, the actual fix turned out to be trivial (Four lines of code) and has since been included in Dokeos.
- MacOS9 users had some issues with uploading documents that did not have a file extension.

From a system administrator's perspective, Claroline was a mixed bag. The admin interface was almost nonexistent. An admin could add or edit a user and some course data, but everything else required editing the database directly. Claroline shipped with PHPMyAdmin, which is an easy to use web front end for the MySQL database, and this was the real administrator interface. Luckily the tables are quite simple to understand even if the column names are in French, but the Help Desk had to refer virtually all Claroline support calls to me since they did not feel comfortable working inside the database directly. Even so, after the initial install, configuration, and rollout the workload dropped off and it took me less than 10 hours a week to administer, fix bugs, and answer user questions.

Usage of Claroline was mixed. Eliminating courses that were automatically created but were obviously not going to have any usage (such as internships and courses with a single enrollee), 83% of courses had at least some professor-added content, even if it was just the required syllabus. 48% sent at least one announcement to the class using that tool. 39% had some content beyond the syllabus, almost always additional documents. The next most used tool, Links, was used in 11% of all courses. Those professors using Claroline beyond the syllabus-only level placed an average of 11 documents and 14 links on their pages and sent 8 announcements.

Students and faculty were surveyed at the end of the year about their opinions of Claroline. The response rate was very low for both surveys (~20%): this was the first year the survey was presented using the portal and many people did not visit the portal in time or did not understand that a survey was available, so these results may be distorted. Some results

- 89% of responding faculty rated Claroline either excellent or good.
- Tools used by more than 50% of the responding faculty were announcements, documents, links, and the user list. No faculty members used the online quiz or chat modules.
- Negative faculty feedback included a poor links tool that did not allow grouping by category and the lack of a way to easily see what the students would see if they visited the site.
- 68% of responding students rated Claroline either excellent or good; another 28% rated it so-so.
- Only 6% of responding students said their professors always used Claroline, but 23% wished they did.
- Most negative student feedback centered around two related topics:
 - "Make the professors use it"

- "Claroline needs to notify you when the professor posts something"

The results were encouraging enough for us to plan to use the product in the following year.

The Dreaded Fork

One disturbing note during our first year of using Claroline was the lack of updates to the software. While the 1.4.2 version we were running functioned well for the most part, I fully agreed with a number of the student and faculty complaints about limitations of the software and it was overly difficult to administer. There were few comments by the programmers on the forums and progress along the roadmap for the next version seemed to have halted. For an Open Source product that is supposed to thrive on frequent releases and upgrades, this was a serious concern.

A cryptic February email from Thomas De Praetere, the lead author of Claroline, left us even more worried. Forum postings and emails eventually cleared up the confusion: Thomas had grown unhappy with the pace of development of Claroline at UCL and had quit to form a commercial company, Dokeos, which would sell a modified version of Claroline as well as support for Dokeos. UCL would continue to develop and use Claroline. While the code bases for Claroline and Dokeos were very similar, Thomas fully expected them to drift apart in the future, making the two incompatible.

A fork of this type is not uncommon in Open Source projects. Since the code is open, major disagreements between developers tend to lead to one developer leaving the project, taking a copy of the code with them, and then modifying it the way that they prefer. One example is the BSD Unixes: the original BSD project has forked into a number of projects such as FreeBSD, NetBSD, and OpenBSD, as well as the commercial Macintosh OSX Darwin. For users this can be a difficult time since frequently one of the projects will attract the majority of developer interest, leaving the other to wither on the vine.

R-MWC eventually decided to move to Dokeos based mostly on the development speed of the project. It was clear that progress toward updated versions was occurring quickly within Dokeos, and over the past year Dokeos has evolved much more rapidly than Claroline.

The Second Year: Dokeos

Despite the name change, Dokeos was still extremely similar to Claroline internally and thus presented few problems when modifying the code to our specifications. Most of the changes from the previous year still worked, and the user interface was similar enough that few faculty felt the need for retraining: 10% of the faculty came to a pre-release demo at the end of the Spring 2003 semester and less than 20% to retraining during workshops before the start of the Fall 2004 semester.

The major Dokeos 1.5 upgrades over the previous Claroline 1.4 code included

- Better notification for students about new or updated materials in classes.
- A much improved document dropbox
- A much improved link tool that allows sorting and categories

- A "Student view" tool for faculty members
- A SCORM lesson import tool and SCORM lesson packager
- LDAP support
- Much better Cascading Style Sheet support
- A much improved admin interface

The improved CSS support has been a mixed blessing. With better CSS, we have been able to make Dokeos appear so much a native part of the portal that few students even realize they are separate products managed by different people, causing some confusion in bug and trouble reports.

In addition, we have added several additional tools directly to Dokeos this past year, including one-click access to the library's electronic reserve system for each class, a drag&drop tool for faculty to upload Smartboard and Starboard save files directly to their class pages and a grade-book module for the first year seminar course which had an exceptionally complex grading structure.

One tool that Dokeos did **not** come with was a course import tool, so that professors who taught the same course year after year would not have to recreate a course web site. The course archive feature of Claroline did not work in the automated course creation environment we use at R-MWC, so I was forced to write a small tool for faculty to import old material. This has become an issue with a number of schools and a "course import, copy and reuse" tool is slated for the 1.6 version.

From an administrative side, this year has been easier. Dokeos is completely stable, with no downtime other than for system upgrades. (Although student reports of Dokeos being unavailable continue. Strangely, these always seem to occur the day before an assignment is due, and don't seem to affect other users accessing the site at the times it was reported as down.) An improved admin interface means many fewer trips into the database, and Dokeos has even stopped including PHPMyAdmin with the package, although any serious administrative use will eventually end up in the database.

Student and faculty survey results for 2004-2005 are not available at the submission date for this paper but will be presented at the conference.

Conclusion

Overall, the introduction of Claroline/Dokeos at R-MWC has been a qualified success. The technical integration of the package into the campus went quite well, and students and faculty have been generally positive about Claroline/Dokeos. The biggest challenge is getting faculty to see the advantages of using the system in ways beyond a simple document storage area. For example, despite R-MWC having a strong focus on global studies, I have only been able to get one faculty member to use the online forum to connect to a school outside the U.S.

The original idea was to test Claroline for three years and see if we wanted to continue with the package or switch to another. Having completed two years, my opinion would be that we will

not switch to a commercial package although moving to Sakai would be a possibility, especially since the next version of our electronic portfolio (OSPI) will require Sakai to run.

Appendix 1: Open Source License Types

The terms Open Source and Free Software are highly loaded among advocates of these philosophies, often spawning some of the most boring flame wars to ever grace the Internet. However, the various software licenses that have evolved out of this debate have important ramifications as far as how people are allowed to use the software.

The Free Software movement was begun by Richard Stallman at MIT in the 1970s. "Free" here does not refer to price, but to the freedom to run, inspect, modify, and redistribute the software--Free software does not have to be cost-free. In order to ensure these rights, he developed the Gnu Public License (GPL), which assures these rights to anyone wanting to use the software while preventing people from using the code in closed source products. The Open Source movement began in the late 1990s as an offshoot of Free Software: Open Source licenses can contain terms that are not permitted under Free Software licenses.

Free and Open Source licenses break down into two major categories, although there are almost innumerable shadings for all of these.

1. GPL-likes. These licenses allow the user to download, run, inspect, modify and redistribute the code with the caveats that the copyright information in the original must remain with the modified version and any software derived from the GPL-licensed original must be released with full source code to the modified version.
2. BSD-likes. The user can download, run, inspect, modify and redistribute the code as with the GPL provided they keep the original copyright information, but they may also modify and incorporate BSD-licensed software into a closed source product without having to make the modified source code available.

A good example of the differences between the two: the early versions of WindowsNT used network code from BSD Unix in a closed source product. This is wholly compatible with the BSD license, but not with the GPL. A college could take a GPL course management system, make substantial changes, and try to sell it, but the terms of the GPL would require the college to make the source code to the modified version available.

Moodle, .LRN and Claroline/Dokeos are licensed under the GPL. Sakai is under the Educational Community License, which allows unrestricted development and is thus more similar to a BSD license.

For more information on both approaches and license issues, see *The Open Source Initiative* and *Philosophy of the GNU Project- Free Software Foundation*.

References

Dokeos. Retrieved 31 Mar 2005 from <http://www.dokeos.com/>

Claroline. Retrieved 31 Mar 2005 from <http://www.claroline.net/>

Sakai. Retrieved 31 Mar 2005 from <http://www.sakaiproject.org/cms/>

Moodle. Retrieved 31 Mar 2005 from <http://moodle.org/>

.LRN. Retrieved 1 Apr 2005 from <http://dotlrn.org/>

Raymond, Eric. The Cathedral and the Bazaar. Retrieved 31 Mar 2005 from <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>

The Open Source Initiative. Retrieved 31 Mar 2005 from <http://www.opensource.org/>

Philosophy of the GNU Project- Free Software Foundation. Retrieved 31 Mar 2005 from <http://www.gnu.org/philosophy/philosophy.html>