

Saturday, May 31. 2008

## **Open Source for Education**

Moodle! Why this Open Source, Course Management System "Can't / Won't / Will Never" Catch On in K-12 Education  
Open Source advocates like to promote the Course Management System (CMS), Moodle!, as the antidote for Blackboard® and WebCT®. These commercial products are "high-priced" and funding for them seems out of reach in a time when school funds are evaporating. Wouldn't it be wonderful if an Open Source product could do the job that Blackboard® and WebCT® do, without the high cost? Unfortunately, the Open Source Course Management System, Moodle! is not up to the task, at least for teachers in K-12 Education. Moodle! may be up to the task in higher-education where the dynamics of course preparation and delivery differ from K-12 Education. For example, a college or university professor may teach a "full time load" of 12 or 15 quarter or semester hours (15 hours is sometimes considered an "Overload"). And, with the rule of three hours of preparation time for each hour of face-to-face class instruction. This schedule results in a full, or overtime load. Public school teachers, on the other hand, spend about six hours a day in front of students, maybe 2.5 times the number of hours that a teacher of a college-level class per week. But, planning time for K-12 teachers is generally 45 minutes or (50 minutes in a generous schedule). So, the numbers look like this;

This table shows that K-12 teachers spend about three times more time in class delivering face-to-face instruction, and K-12 teachers receive one-tenth the preparation time.

These data demonstrate the reason that a Course Management System for K-12 teachers must have a streamlined interface and be optimized for time-saving. A complicated interface and tedious course development and management process fail to meet the needs of K-12 teachers and are not viable course management and instructional delivery solutions. What are the Problems? Moodle! does what it says it will do, i.e., manage online courses. So, what are the problems that affect ordinary teachers? The first problem stems from the "database backend-handcuffing" by the product. The backend database interface is slow and tedious. Instead of being able to use the File Transfer Protocol (FTP) system that would automate sending the course updates to the online server, each piece of instructional material (title, description, activity, image, graph, video link, etc) has to be copied and pasted into an online database form. This backend database "adding and updating" method prevents teachers from using cute and time-saving tools such as "global search and replace." The backend database "copy and paste" Moodle! method of course management also prevents the streamlined use of templates. Moodle! also lacks a checkout feature. Moodle! A checkout feature means that when someone else is working on the course materials, you cannot work on that same component, too. This prevents one person on a development team from overwriting the work of another. This feature is vital for collaborative instructional design and team-teaching.

Another Moodle! shortcoming is a course update feature that depends on a crude "backup /restore" strategy. This is a "full database replacement" strategy that allows only for the restore of the entire course database. This means that you cannot update just a chapter, section or subsection of a course, but this means that you must backup the entire course when you want to update in this way. The only way to add section and subsection content in a Moodle! course is to "copy and paste" the new content (title, description, text, image, link, activity, etc.); one-by-one, bit-by bit into the online database. Or, teachers can work on a complete copy of the database, and replace the online course with a complete, new copy of the course. The problem with this strategy is that once data (such as student records, student contributions, student grades) are backed up, the old records are copied over (over written) and no longer exist. Therefore, the course update process only works correctly before students start to use the online course. Desktop Development Environment and Development Servers Moodle! provides a "desktop development environment" that works with Microsoft® Windows. This is useful for developing and prototyping a course. What this means is that teachers must have a complete working database on their desktop, and they must copy the entire database to the server. But team teachers cannot have more than one copy of the full database on their desktop computer unless they are able to share that desktop somehow. Moodle! Caution: Do not attempt to share the Moodle! Development Environment (MDE) on a school district network. The MDE lacks any of the basic security features that members of your school district's IT Department staff demand. What this means is that teacher teams cannot develop courses that fit together. The method for more than one teacher to work together is to use the live, online database. Fortunately, Moodle! has a way of allowing courses that are in the development stage to be hidden from the view of students. Moodle! What teachers have to do if more than one person is working on a course is to assign (delegate, deputize, shaft) one person on the team with the job of entering everyone's course contributions. Of course this requires double work for one person. And, as our graph shows, teachers do not have time for double work. What Moodle! needs is the ability for multiple members of a teaching team to update courses in a modular way, rather than its current "all or nothing" database strategy. Moodle! Multiple Workstations - Multiple Team Teacher: Moodle! Catastrophe The lack of a modular course management structure creates another course

management nightmare for Moodle! Unfortunately, teachers must use computers at work and at home, and keeping a database synchronized is next to impossible. The only solution to this issue seems to be to develop Moodle! in separate courses on a development server, then copy the separate courses to the production server once the course is completed. The alternative, if each teaching team member develops course with the individual desktop development environment, is to "copy and paste" each individual component and send those components to one of the team members; the one that will copy and paste the individual components to the active server. This strategy represents "triple work for one team member; and will never get done!"

But, in these days of "test-stress-driven" curriculum objectives and minimum preparation time, teachers barely have time for their first level of work, i.e., face-to-face instruction of students; let alone course development work in triplicate.

Consequently, "gung-ho techie teachers" are the only folks who might manage a K-12 Moodle! course after the first, time-intensive week. "Mulish" Lack of Flexibility Teachers must be able to save and reuse course components and templates when developing courses. Flexibility to "mix and match" course components, and flexibility to share components across-courses must be available. But, these are missing in "mulish Moodle!". The requirement of "backup and restore" of the entire database hampers efficient development. Just keeping database copies in synchronization is next to impossible by a single person, let alone multiple teachers. Of course, with college classes where the professor prepares the entire course syllabus in advance (Remember Higher-Ed folks have ten times more planning and course development time as K-12 teachers), Moodle! functions quite effectively. But, college courses are packaged into neat quarter or semester packages with a limited number of class meetings. Compare sixteen sessions in a Tuesday-Thursday semester course with the 90 sessions that K-12 teachers meet with their classes per semester.

Adventurous (or masochistic) teachers must break courses into "units." But even so, these units will be "supplemental materials, not core course components. Desktop Development Security: A System that can't Connect to the Internet can't Develop an Online Course As mentioned, Moodle! provides a local desktop development platform. Unfortunately, this development environment lacks basic network security, and cannot be used on a network such as the kind of network that school districts employ. Teachers cannot connect a computer with this MDE installed to the Internet from home, either, especially with an "always on" broadband connection (DSL or Cable). Besides the local Windows®; Desktop Development System (that most teachers require), creates that same dual database issue. What happens when the teacher wants to work on the desktop at school as well as at home is "confusion." Needed Usability Features What Moodle! needs is a front end interface similar to the "Dreamweaver®;/ Contribute®; system, with version control and checkout components. Updates to courses should be "one-button, FTP-enabled." And, Moodle! course development needs to be both "cut and paste" and "drag and drop."

The software needs to work with templates and modules, both used as "course building blocks." Building every course from scratch is an effort in futility that few teachers have time for. Instructional Inadequacies Instructional planning and course design can be streamlined by using templates, modular elements and reusable forms. Moodle! needs to create this flexibility. Copying entire courses using a "backup and restore" function is equivalent to repeating every minute (starting again from the beginning) of class instruction each time the class meets.

Besides, several teachers, building components of a course would still need to delegate one person to copy (integrate) their contributed components into the final version. Double or triple work that every teacher abhors. Practical, Logistical Issues There were other practical and logistical issues associated with deploying a Moodle! course:

For example a Fourth Grade teacher with 22 students, teaching seven classes a day... Reading

English and Spelling

Math

Science

Social Studies

Health

... would require an inordinate amount of course development time. But, Moodle! course development is too time-intensive for teachers to manage for so many classes. Besides that, every student must have complete access to a computer for every class. Otherwise, only a select few students could use the course at one time. And, only a few students being able to profit from the extended time that the course development takes makes course development impractical. Sidebar Teachers can only afford to automate instructional components when these multiply the effect of their work. Any process that gouges into teachers' limited planning time cannot be justified because such methods squander time, a teacher's most precious resource. So what about students accessing the materials at home for review?

Posted by Classroom Toolkit Newsletter in Open Source at 01:00

Just a few comments ..

The chart and graph in comparing K12 to Higher Education is based upon what data?

It's reasoned well (by the author to support the message of the article), but makes no reference to true data collected in any research that compares teaching in K12 to that of Higher Ed.

"Moodle is the problem" with it's "database backend-handcuffing".  
No "global search and replace".

Moodle does have global search and replace where needed - just not what the author appears to be accustomed to ... DreamWeaver

like. How many teachers can actually use a DreamWeaver-like product and it's FTP button to provide: forums and discussion boards, quizzes, and individual student Journals to name a few Moodle tools.

No "checkout".  
NOT needed.

Restricted to uploading files one at a time.  
Simply NOT true. A student assignment, for example, can allow multiple file upload.  
A Teacher could learn to use ZIP to create an archive of all files desired to upload, upload the zip, then un-zip it. That's not a hard process for any Teacher or Student to learn (and should learn a "time saving" skill which the author seems to tout so much).

Check out NOT needed.

Course update

There is a process for course updating, true. Hmm, to prepare a course for next semester, it's a very simple process ... which follows this basic outline:

backup current course with user data

download the backup for archival purposes

In the admin menu of course, click the "Reset" link and choose what items to clear from the course (users and files, assignments, etc.).

Done!

Course is now ready for new student enrollments.

I find your evaluation of Moodle, biased and lacking in knowledge of the tool and therefore invalid. Your readers (if they read this at all) should also be allowed to read this comment and encouraged to seek advice and do research else where.

Anonymous on Aug 20 2009, 00:53