Course Management Systems:
Time for Users to Get What They Need

By Andri Ioannou and Robert D. Hannafin

Course management systems (CMSs) are software systems designed to manage course content and course activities. These tools (e.g., WebCT, Blackboard, ANGEL, Educator, FirstClass) integrate technological and pedagogical features into a web-based system that allows instructors, even those who are unfamiliar with web-based technologies, to design, deliver, and manage an online course. Common features of most CMSs include content areas, discussion boards, chat rooms, assignment drop boxes, quizzes and surveys, and white boards. CMSs support student-teacher communication and collaboration; students are able to share resources, collaborate, participate in forums, take online tests, access their grades, and upload assignments.

Today, CMSs support thousands of courses at colleges and universities and that number is growing at a staggering rate. CMSs are used both to support and supplement face-to-face instruction, a so-called "blended" approach, and to deliver fully online web-based courses. Osguthorpe and Graham (2003) defined blended learning as an approach that combines face-to-face instruction with distance education delivery systems. Blended courses use a CMS to support and enrich a variety of instructional needs, tasks, and experiences. Those who use a blended learning approach, according to Osguthorpe and Graham (2003), are trying to take full advantage of the benefits of both face-to-face and online instruction to use "the web for what it does best and class time for what it does best" (p. 227). Fully online courses, on the other hand, meet exclusively in the CMS. Students access all course materials (lectures, assignments, resources, etc.) and communicate via email, discussions, and/or chats with the instructor and other students. Frequently instructors' scheduled office hours become virtual via online synchronous communication.

While CMSs were initially developed to support distance education and fully online course delivery, they are now used predominantly in on-campus classroom settings to complement traditional courses (Harrington, Gordon, & Schibik, 2004; Morgan, 2003). The ease with which users can organize asynchronous and synchronous communication activities (e.g., chat, discussion) in CMSs is one of its most powerful features because it enables (in fact, arguably, encourages) instructors to create and support dynamic learning communities consistent with a social constructivist perspective. This certainly explains some of the growth in on-campus blended courses (Dabbagh, 2004; Morgan, 2003).

Perhaps no other innovation in higher education has resulted in such rapid and widespread use as the CMS (Harrington, Gordon, & Schibik, 2004). By 2002, over three-quarters of all colleges and universities in the U.S. had adopted a CMS, and nearly one-fifth of college courses used a CMS (Campus Computing Project, 2002). By 2004, just a few short years from their introduction into higher education, such systems could be considered ubiquitous on college campuses (Molenda & Bichelmeyer, 2005). The recent merger of Blackboard and WebCT made Blackboard, Inc. the world's leading provider of integrated e-learning systems, by itself serving more than 3,700 academic clients in more than 70 countries worldwide (Blackboard.com, 2005).

It is possible that CMSs have introduced so much new functionality so fast that end users have not had a chance to seriously reflect on

what they need or want. Perhaps users are understandably giddy about the instructional tools they now have at their fingertips and have not yet critically examined these tools to ask "what else do I need?" How can these tools get even better to satisfy educational expectations and needs? In the following paragraphs we review some literature to identify common weakness across CMSs. We then recommend technological accommodations that should make learning with these tools more efficient and enjoyable.

CMS deficiencies

While CMSs have become more accessible and increasingly useful, many instructors and learners in higher education have expressed frustration over things like response time and ease of use. Morgan (2003) sampled 880 faculty and instructional staff at a large Midwestern university and found that 80% of CMS use was to enhance regularly scheduled face-to-face classes. She also interviewed the 5% of the faculty who had reported decreased CMS use since they first started using it. These faculty members indicated that time spent and difficulty of use were the main reasons for this decrease. Some faculty complained about the time required to load instructional materials in the course area, and others found the CMS inflexible and overly structured. Morgan concluded that CMSs must get easier to use if they are to be accessible to all users.

Siemens (2004) examined the drawbacks of CMSs and suggested functionalities that a CMS needs to acquire to meet user needs. The author asserted that CMS interface is confusing to many users, and needs to be simplified. Current CMS interfaces rely on "what do the designers/administrators want/need to do, [rather than on] what the end users want/need to do" (Siemens, 2004, "Drawbacks to Learning Management Systems" section).

In a recent study, Jafari, Mcgee, and Carmean (2006) investigated the advantages and limitations of CMSs through interviews with 59 faculty, 52 students, and 51 academic, library, IT administrators, and IT service providers from seven campuses. These institutions were using one or more CMSs including Blackboard, WebCT, eCollege, and ANGEL Learning. All users expressed dissatisfaction with the speed, efficiency, and intuitiveness of current CMSs. Administrators valued the potential of smart error messages in terms of relieving frustration and time for both faculty and support staff: "Smart metatagging and searching could make accessing files and finding threads more efficient" (p. 4). Faculty members wanted user-friendly functions that required fewer clicks to make tasks such as entering grades or returning files less time consuming. They would also like to have more feedback, warning, and notification mechanisms within the system. However, the most frustrated end users were the learners who generally characterized the current CMSs as "dull." The authors observed:

> For all the slow, backward, and clunkiness [sic] of the current systems, students noted that the environment isn't even easy to figure out: the navigation is confusing and takes too many clicks, assignments get lost, the discussion can't be sorted with newest on top, the system doesn't learn anything about the learner and never sends reminders or status messages (p. 8).

The need for more efficient and intuitive CMSs is obvious; thus, future studies need to focus on CMS technological drawbacks. CMS users should continue to critically evaluate the current tools to determine what accommodations might improve their online experiences. The authors explored some of the widely used CMSs and recommend that a client site application can improve both the effectiveness and efficiency of learning delivered by CMSs. A short discussion of web-based versus client applications follows.

Web-based versus client applications

There are two approaches to the development of a new software application; developers can choose to build it for the web or to build a "rich" client (desktop) application to run on personal machines. Webmail is a popular web application that allows users to read and write e-mail on the Internet through a web interface. Examples of Webmail applications are Hotmail, Yahoo mail, and Gmail. Notable rich clients (desktop applications) are email clients such as Microsoft Outlook and Eudora, synchronous communication applications such as Instant Messengers and Skype, and authoring tools such as Macromedia Dreamweaver and Flash.

There are pros and cons associated with each implementation. Olivier (2004) argued that web applications are incomparable at finding, retrieving, and sharing information, but not as

"It is possible that CMSs have introduced so much new functionality so fast that end users have not had a chance to seriously reflect on what they need or want."

effective for authoring purposes. On the other hand, "desktop tools are generally easier to use, more responsive and, with local storage, allow the user to continue to work in disconnected mode" (Olivier, 2004, p. 2). Spolsky (2004) argued that web applications involve lower user administration costs, provide higher security, and are easier to deploy because there is no installation. However, he asserted that rich client applications offer faster response time and enable much more interesting user interfaces. Spolsky (2004) went on to provide a few examples of the advantages of client applications that cannot be replicated in a web application:

> *"CMS users should continue to critically evaluate the current tools to determine what accommodations might improve their online experiences."*

1. Fast-drawing program (web interfaces are generally slow)
2. Real-time spell checker with wavy red underlines (not available in web-based applications)
3. Warnings that users will lose unsaved work
4. Fast keyboard-driven interface that doesn't require a mouse (e.g. CTRL+F: open the Find and Replace window)

We explored the most recent versions of widely used CMSs including: Blackboard 6.2 Enterprise, WebCT Vista 4, Sakai 2, ANGEL 6.3, Desire2Learn 7.3, Virtual-U 2.5, ATutor 1.5, Learnwise and The Learning Manager 3.2 (*http://www.edutools.info/*, *http://www.cmsmatrix.org/*). They are all web-based, meaning that the course package is accessible via a conventional web browser over the network (Internet or intranet), and have no client-side software to install. The drawbacks of web applications, described by Olivier (2004) and Spolsky (2004), are consistent with the technological concerns associated with CMSs, described by Morgan (2003), Siemens (2004) and Jafari, Mcgee and Carmean (2006). But, how much response time, user friendliness and ease of use are end users (instructors and students) willing to sacrifice in exchange for the formidable benefits of web-based applications?

Benefits of adopting client-site software

We propose that CMSs should move to client/server architecture and provide client-site software to be installed locally on users' personal computers. This would make the course package accessible either via the CMS desktop application or via a web browser, based on user choice. Although not an exact analogy, accessing the course package via the CMS desktop application would be similar to accessing your Yahoo-mail using Outlook. Everyday, many users open their browser to access their Webmail, while others prefer to check their email using a rich client application with a consistent interface, customized to the user's look and feel. According to Olivier (2004), most people in fact access their email on the web (using Webmail) only when traveling or when no other alternative is available. Most people still prefer the features afforded by working with desktop applications (Olivier, 2004). Surely part of the popularity of desktop applications like email clients, Instant Messengers, MS Word, or Macromedia Dreamweaver is their user-friendliness, customizability, and efficiency.

We are not suggesting new pedagogical approaches to learning supported by CMSs, but rather technological accommodations that should make CMSs more useful and efficient, thus increasing student and instructor satisfaction and improving the overall learning experience.

Benefit 1: Clients are faster.

By relying on the local processor and by using local storage, client applications offer faster response time (Olivier, 2004; Spolsky, 2004). Instructors' complaints about time required to load instructional materials (Morgan, 2003), or students' frustrations with the slowness of the environment (Jafari, Mcgee, & Carmean, 2006), would be diminished with a client application. In addition, by running background "threads" client applications allow users to multitask and be more productive. Background threads allow essential tasks to be done judiciously so as not to impact the user experience. Tasks such as auto-saving files in MS Word, auto-recovering data if MS Excel closes unexpectedly, or checking for updates on the server in Outlook every ten minutes, are implemented through background threads.

Moreover, client software would speed up other procedures. For instance, all of the CMSs we reviewed require user authentication to grant access to the course area. The login action (typing username and password) is time consuming, especially if the user accesses the course multiple times a day. The more extensively a CMS is used in a course (e.g., online discussions, online tests, assignment delivery), the more often instructors and learners need to login on the course area. This is particularly wasteful in cases where there is nothing new posted that requires the learner's attention.

Benefit 2: Clients are more efficient.

Unlike the problem where users repeatedly logon and find no news, there is nothing preventing users from failing to logon when there is something that needs attention. A CMS desktop application would enable "transparent" access to the course area. Users can work on other tasks on their personal computers as their CMS client software runs in the background in communication with the server. In this circumstance, as with other client applications for email or instant messaging, new events, including postings in the discussion board, calendar/syllabus updates, and who is available online, would be announced to the users right away in the form of a pop-up window/message, a specific sound, or an icon appearing on the taskbar. Currently, some CMSs offer dynamic notification or automatic email notification of the most recent changes in the course environment; however, this assumes that users will login on the CMS or check their email (internal or external email). So while this feature is helpful, it does not meet the end users' needs (Jafari, Mcgee, & Carmean, 2006).

Benefit 3: Clients are more user-friendly.

Difficult and confusing interfaces distract learners from what they need to focus on. The ease of use and the uniformity of the interface may make possible a higher level of concentration on the material being studied (Edwards & Holland, 1992). Desktop tools are generally easier to use and learn (Olivier, 2004) because they employ windowing features, menus and task bars (toolbars), and meaningful icon buttons that are easy to memorize. Moreover, in a desktop application users can enable real-time spelling and grammar tools when they craft a posting (Spolsky, 2004), sort messages by date (Jafari, Mcgee, & Carmean, 2006), drag-and-drop, undo-redo, and access dynamic help (e.g., Microsoft Office Assistant) – all of which are functions of popular client applications such as MS Word.

Benefit 4: Clients can be personalized.

Dix, Finlay, Abowd and Beale (1993) argued that it is not enough to have a system accessible for all users; users should have *control* over the tool and be given the opportunity to impose their own structure on the way information is presented. In existing CMSs, instructors have some control over the interface of their pages (ability to change the graphics and the way a course looks), while learners have none. One of the biggest benefits of rich client applications is the ability for end users to change their look-and-feel (Spolsky, 2004). CMS client software allows user-custom-

ization to fit users' individual needs. For example, users can change themes, font size, or icons. Also, they can set up and use a keyboard-driven interface (otherwise hot keys) for their most frequent action, currently unavailable web applications (Spolsky, 2004). Such functionality simplifies navigation and reduces the number of clicks to complete a task (Jafari, Mcgee, & Carmean, 2006). Finally, users can manage the overall activity of a desktop tool and impose their preferences, for instance, to set it up to start automatically when the operating system loads, check for new updates every five minutes, and save work automatically every three minutes.

Conclusions

This article has commented on several technological issues related to CMSs, and argues that CMSs should adopt client-site software to make learning more efficient and enjoyable, and possibly even more effective. Users are not always looking for the many features that vendors seek to implement; users value *ease* and *speed*, both characteristics of a client application. Technology's affordances and limitations highly affect the quality of the instruction (Kozma, 1994) and even in some respects, limit the vision of the instructor and learners. We encourage all users to critically evaluate these tools to determine what accommodations might improve their online experiences. Users, not vendors and markets, should guide the new generation of CMSs.

"It is not enough to have a system accessible for all users; users should have control over the tool and be given the opportunity to impose their own structure on the way information is presented."

Andri Ioannou *is a doctoral student in the educational technology program in the Department of Educational Psychology at the University of Connecticut. She received a B.A. in computer science from the University of Cyprus. Her research interests include web-based learning, online learning environments and multimedia technologies in education.*

Robert Hannafin *is an associate professor in the Department of Educational Psychology at the University of Connecticut. He received his Ph.D. in learning and instructional technology at Arizona State University in 1994. His research interests include technology enhanced learning environments and technology integration in K-12 and teacher education.*

References

Dabbagh, N. (2004). Distance learning: Emerging pedagogical issues and learning designs. *Quarterly Review of Distance Education*, 5(1), 37-49.

Dix, A., Finlay, J., Abowd G., & Beale, R. (1993). *Human-computer interaction*. Hillsdale, NJ: Prentice Hall.

Edwards, A. D., & Holland, S. (1992). *Multimedia interface design in education*. Aberdeen, U.K.: University of Aberdeen.

Harrington, C. F., Gordon, S. A., & Schibik, T. J. (2004). Course Management System utilization and implications for practice: A national survey of department chairpersons. *Online Journal of Distance Learning Administration*, 7(4). Retrieved September 4, 2006, from http://www.westga.edu/~distance/ojdla/articles/winter2004/harrington74.htm

Jafari, A., McGee, P., & Carmean, C., (2006). Managing courses, defining learning: What faculty, students, and administrators want. *EDUCAUSE Review*, 41(4), 50–71. Retrieved September 4, 2006, from http://www.educause.edu/apps/er/erm06/erm0643.asp

Kozma, R. B. (1994). Will media influence learning? Reframing the debate. *Educational Technology Research and Development, 42*(2), 7-19.

Molenda, M. & Bichelmeyer, B. (2005). Issues and trends in instructional technology: Slow growth as economy recovers. In M. Orey, J. McClendon & R. M. Branch (Eds.), *Educational media and technology yearbook 2005* (Vol. 30) (pp. 3-28). Englewood, CO: Libraries Unlimited.

Morgan, G. (2003). *Faculty use of Course Management Systems.* (Technical Report No. ERS0302). Boulder, CO: EDUCAUSE Center for Applied Research. Retrieved September 4, 2006, from http://www.educause.edu/ir/library/pdf/ERS0302/ekf0302.pdf

Olivier, B. (2004). *Application tool component framework*. Retrieved September 4, 2006, from http://www.elearning.ac.uk/resources/

Osguthorpe, R. T., & Graham, C. R. (2003). Blended learning environments: Definitions and directions. *The Quarterly Review of Distance Education*, 4(3), 227-233.

Siemens, G. (2004). *Learning Management Systems: The wrong place to start learning.* Retrieved September 4, 2006, from http://www.elearnspace.org/Articles/lms.htm

Spolsky, J. (2004). *How Microsoft lost the API war*. Retrieved September 4, 2006, from http://www.joelonsoftware.com/articles/APIWar.html