

Course Management Systems: Expectations and Outcomes

**Debra T. Burhans¹, Carl Alphonc², Helene Kershner³, Barbara Sherman⁴,
Deborah Walters⁵**
¹Canisius College/^{2,3,4,5}University at Buffalo (SUNY)

In the 1999-2000 academic year, an ambitious overhaul of a large, introductory computer literacy course at the University at Buffalo (SUNY) was initiated with the help of a grant from the Pew Learning and Technology Program¹. This course enrolls approximately 1200 students per year, divided into lecture sections of 200 and laboratory sections of 20-40 students. With the help of on-line and LAN-based technologies for teaching, testing, and course management, we transformed the course from a traditional lecture format with short, unstructured laboratory meetings to a more student-centered model with fewer lectures and more structured laboratory experiences. While this course is not specifically targeted towards science and engineering students, the issues addressed herein are of interest to all educators who use or are considering using a course management system. Faculty for the course are technology-savvy computer scientists whose familiarity with computing technology influences their expectations regarding content and performance of off-the-shelf educational technology products. Specifically, expectations with regard to the course included the following:

- A course management system (CMS) would decrease time spent on course administration.
- Grading time would be reduced by shifting to on-line and web-based training and assessment tools. The CMS would facilitate on-line grade maintenance.
- Lecture time would decrease due to the use of interactive, on-line tutorials for topics that have proven difficult to cover in lecture. These interactive materials would be integrated into the CMS.
- As a result of the above-mentioned technologies, there would be a cost savings in faculty time and lab assistant expenses that could be recouped by the department (dollars) and by the faculty members involved in the course (more time for research).

There were positive outcomes related to the incorporation of technology materials into the course, including a measurable increase in student learning and satisfaction as well as cost savings for the department. However, there were also a number of negative outcomes. Administrative time spent on the course increased considerably, and grading time did not decrease. In addition, troubleshooting computer-related glitches, including problems such as network bottlenecks, software performance, and inability of different pieces of software to communicate with one another, required much more faculty time than anticipated. The expectation that the CMS would allow for coordination and integration of different on-line resources was not realized. While there have been a number of studies of the impact of technology on student learning and outcomes, our focus is on the faculty perspective of teaching with technology. Our experiences using on-line materials including a CMS in a large undergraduate course have led to the identification of a number of factors that influence the effectiveness of computer-based educational technologies. These findings should prove valuable

to other educators, administrators, and vendors involved in developing and implementing these technologies.

Course Administration and CMSs

In the best case, a CMS serves as a central repository for electronic resources associated with a course, including pedagogical and administrative materials. Through a CMS students should be able to access on-line learning materials, grades, assignments, syllabi, and course schedules. The CMS should provide information about asynchronous access to instructional staff and project group members such as email addresses (which should be given as links that directly invoke an email program) and office phone numbers. In addition, synchronous communication facilities such as chat should be available. Instructors should be able to manage an entire course within the CMS, with easily employed utilities for integrating a variety of on-line materials. They should be provided access to students and groups of students via email and synchronous chat.

As early users of a newly installed CMS, we discovered a number of problematic issues that seemed to have not been anticipated by either the CMS developer (in this case, Blackboard Systems) or the instructional technology staff at our university. Problems ranged from a lack of control over student accounts to the inability of the underlying computer system to handle a large-enrollment course with multiple assignments submitted electronically. As a result, the use of a CMS for course administration involved significantly more time rather than less when compared to previous semesters. The following is a list of desiderata for CMSs based on our experiences.

A CMS should interact with the campus-wide computer services so that students have the same user name and password for the CMS as for their personal campus accounts. Absent this, either the professor or a CMS administrator will have to cope with problems of students forgetting passwords and user names, which is extremely time consuming. It is extremely frustrating for students to have different passwords and/or user names for different computer systems on campus. In our case, directing students toward different information technology (IT) administrators and troubleshooting problems of student access was a burden.

A further problem with passwords and access concerns the integration of proprietary on-line materials into a CMS. Such materials generally come with their own access codes, login names, and passwords. In case students select their own passwords and/or user names, these are often forgotten, and the instructor (who frequently cannot set passwords or user names for such materials) must track down an administrator who can set things aright. The time cost of such activities should not be underestimated. One of the goals of the Pew Learning and Technology Program was for instructors to integrate materials developed elsewhere into their courses rather than spend time developing their own materials. Such development is costly in terms of time and money, and requires an ongoing commitment on the part of faculty to upgrade and revise these materials. A solution to this problem might involve incorporating access privileges to externally-hosted proprietary materials into the CMS, where a particular student account might be set up to include all necessary access information in a manner that is transparent to the student. Such a solution requires cooperation between publishers, software developers, campus information technology, and may be too difficult to coordinate. Note that professors who develop their own software avoid some of these problems.

A CMS should be flexible in terms of uploading and downloading grades from other programs in a variety of formats. There are two aspects important in grading. First, grades come from multiple sources in a variety of formats. Some may be in plain text files, while others may be stored in spreadsheets. A university computing center may provide scoring services for tests that employ scan-able forms, and the data may be returned to a professor in a variety of formats. It is critical to be able to merge grades from multiple sources into a single grade file. While grade management is not a focus of CMSs, it represents a time-consuming task that is clearly a component of course administration. Second, once grades are stored in a CMS, it should be possible to change weightings and to scale points for assignments. If this capability is not present in the CMS, the CMS should be capable of exporting grade files to spreadsheets and importing grades from spreadsheets, where a spreadsheet can provide the full range of options for grade adjustment. CMSs generally provide nice interfaces to grades once they are stored, but the capability for grade merging and management is not a primary focus of CMSs. Absent a “grade grinder” program that might merge grades in various formats together, we continued to hand-enter grades into a spreadsheet. Due to the inability of the CMS we used to upload grades from a spreadsheet, we posted grades on a web page rather than using the built-in grade utility, copying and sorting only those parts of the grade spreadsheet that did not reveal a student’s identity. This process can be automated with a macro, but nevertheless is time consuming, and is neither as effective nor as private as utilizing the CMSs ability to display grades for a single student who has authenticated him or her self to the system.

A CMS should be appropriate for the course size and content: it is best verified in advance that a large-enrollment course with many assignments to be turned in electronically will not stress the capacity of the CMS or the underlying system. Our first semester of using a CMS resulted in a complete system crash when close to 1000 assignments were electronically submitted. When the underlying computer hardware was enhanced the system remained too slow to make on-line retrieval and examination of assignments workable. We were more satisfied with a submission program that was developed in our department, though it lacked a good interface for grading.

A CMS should provide an easy way to provide students with feedback on assignments. When an assignment is being graded, there should be no need to find the name of the student who submitted the assignment in a list in order to email him or her feedback: there should be a simple, direct link to the student's email. After all, the CMS “knows” which student submitted the assignment. In our case, selecting a student name from a list of over two hundred in order to initiate a feedback email made the process impossible.

A CMS should provide an easy mechanism for grouping students, teaching assistants, and instructors in a dynamic fashion, so that students can be members of more than one group and group membership can change over time. We were unfortunately not able to take advantage of the grouping features in our CMS because they did not provide the flexibility we needed: to sort students in a number of different ways and to move students between groups. The grouping feature should allow for the specification of a variety of criteria for groups, for example, select students with a particular grade in the course at a certain date.

The Mismatch Between CMS Performance and Expectations

One of the frustrations of teaching a large class is that as much or more time is spent on administration of the course as is spent on developing innovative course materials. Our hope was that a CMS would free us from many of the tedious tasks involved in administering a large course. Some of our expectations are related to the fact that our discipline requires student projects: we can not simply use exams to test knowledge. Even on-line, animated testing software does not test the ability of a student to see a project through from beginning to end. Thus, we need to be able to integrate grade data from a variety of sources, including email from teaching assistants.

Successful users of CMSs often have small courses or courses that employ standardized testing and do not have significant project components. In some cases they have been able to develop their own application software either for enhancing an existing CMS or they have created their own CMS. Enhancing a CMS would be time consuming and the programs developed to this end would require maintenance over time, which we have already noted is costly and time-consuming.

The continuing development of innovative tools to help students master concepts in courses and to enable students to interact with one another on projects is an exciting area of research, and an important advancement that is critical to improving and furthering on-line education. However, tools that ease the daily grind should not be ignored.

An additional problem of using a CMS is not related to the software itself, rather, the vision of its use as put forward by university and system administrators. This leads to frustration when the faculty are themselves experts in the area of computing. Faculty for our course are well able to design web sites and are capable of writing utility programs to perform tasks such as reformatting files. In fact, we are accustomed to having considerable control over the production and presentation of our on-line materials. Thus, where we might be capable of designing and implementing a program to format some of our existing materials so that they would work with our CMS, instead we are not allowed to “tinker” with the system at that level. This is reasonable, considering the dependence upon the CMS of a large portion of the university community. However, this does not alleviate the frustration experienced in this regard.

CMSs and Sharing of Course Materials

One of the most exciting uses of the Internet for teachers is the ability to look at the ideas of other teachers. Reviews of books, syllabi, innovative projects, etc. are freely available to peruse, leading to new ideas. Unfortunately, the move on the part of university administrators to create an electronic component for most courses has involved the use of CMSs, which hide the wealth of teaching information we had only begun to appreciate. Thus, it is important that university professors and administrators provide a mechanism for allowing guest logins to a CMS whenever possible or appropriate. Obviously, a CMS may contain proprietary information that can not be shared with the general public. However, working to make as much information as possible available is an important goal. Aside from the fact that CMSs render course information inaccessible, the information therein may never be discovered in a Web search, even if some form of guest login is available.

Grading and Testing

One of the promises of electronic course materials is that grading can be incorporated into on-line exercises, obviating the need for human graders. In our course there are exams, quizzes, on-line labs and exercises, and homework/projects. The exams are given on paper scan-able forms. Electronic quizzes are problematic due to the fact that there are more than 20 lab sections, all meeting on different days at different times. Coming up with a database of quiz questions that was sufficiently large proved impossible. Thus, paper quizzes are given in lecture and hand-graded. On-line labs and exercises are for the most part not fully incorporated into the CMS, and are accessed from Web sites or CD-ROMs. The grades derived from the on-line exercises are generally emailed to a professor in text form. Projects are submitted on-line but graded by hand. The following list addresses important issues in on-line grading and testing.

In order to use on-line testing, it is critical to have an appropriate testing facility. Further, some types of courses are more appropriate for on-line testing than others. These two points are intertwined: if a large, good-quality electronic test bank can be developed, then students in a course need not all take an examination at the same time. This means that the size of the testing facility may be much smaller than the size of the class. On the other hand, if it is difficult to develop a large number of good questions, then large numbers of students may need to take an on-line test simultaneously. This requires a very large on-campus laboratory that can be made available for testing when needed. A particularly successful example of the use of a large laboratory using on-line course materials with on-line testing is the Math Emporium at Virginia Technical University².

Our experience points to differences in the ease of creating large question banks for on-line testing. There are many numerical types of problems that can be infinitely varied by simply changing the numbers. Rather than specific questions, general problem forms can be defined and instantiated. Our experience with computer literacy is that the number of good questions that test a particular topic are limited. Despite the fact that we had a team of experienced teachers in the course, we found it nearly impossible to create enough questions to allow for more than a few versions of an examination. Textbooks provide test banks, but the quality is not necessarily good, and again, the number of questions is still limited when a course with hundreds of students and tens of lab sections is considered. Thus, if on-line testing is to be employed, either a large number of questions of equal difficulty must be available to choose from, allowing for the creation of many different test versions, or a large computing facility must be available so that most of the students can take an examination at the same time. The issues for tests are clearly the same for quizzes, though the stakes are generally lower for a quiz and duplicate questions may be less of a concern.

Finally, most of us have many tests and quizzes from previous semesters in a variety of formats, perhaps as plain text, latex, various versions of Microsoft Word, or Word Perfect, etc. Despite the plethora of formats, there is generally a mechanism for converting files to plain text. It would be extremely helpful if there were a "test grinder" of sorts that could read a file and put it into a format compatible with the CMS testing facility. Entering tests and quizzes into a CMS by hand is extremely tedious and time consuming: far more time consuming than simply creating a test in a word processor.

The Lack of Integration of Materials

There are some excellent teaching materials available on the Web and on CD-ROM. However, there are issues to be considered if such materials generate grades for students. If they are separate from the CMS, how are student grades forwarded to a professor? How do students identify themselves? We wrote a C++ program to parse the grades emailed from on-line learning materials, but encountered problems. Students did not necessarily identify themselves using their full names, so matching sometimes had to be done by hand. Students typed identifying information incorrectly, such as student ID numbers. The emailed grade files were not completely secure: a clever student could have “faked” grades for the exercises. Finally, the output of the program still had to be hand-entered into a spreadsheet. This is just one example of the way in which different materials are available, yet they aren’t coordinated with one another, leading to additional work for an instructor and his or her teaching assistants.

In our quest for on-line teaching materials we initially thought we could use an on-line tutorial to teach students about various productivity software products, such as Microsoft Office applications. Such tutorials prompt students to perform different tasks, then score the students’ efforts. The problem of merging grades from separate sources that was mentioned above remains. However, we encountered another unexpected problem: we had underestimated the value of having a student complete a project that integrated different applications from start to finish. For projects like this there is, as yet, no “automatic” grading program. In fact, the problem of developing such a program seems as complex as tackling the problem of natural language understanding. Such projects, thus, are still graded by hand, though they can be submitted on-line. Thus, “old-fashioned” assignments may be important, and may prove difficult to grade electronically.

Lectures

One of the advantages of shifting to on-line materials is that some topics are difficult to cover in lecture: they are best left to students to discover, particularly when there is an interactive environment in which to do so. The first semester that on-line materials were available we reduced the number of weekly 50-minute lectures from three to two. Student feedback at the end of the semester indicated that they missed the third lecture and thought it should be restored. This occurred despite the fact that we had added an additional two hours to the laboratory meeting times. The second semester we added the third lecture back but made it optional. Instead of a traditional lecture, it was offered in a large computer laboratory classroom where students were seated in front of computers. Approximately half of the students attended the optional lecture regularly. This was important, as the number of available computers was approximately 25% of the total lecture enrollment. This lecture covered the use of on-line materials, including on-line labs and other course software. In addition to the instructor, several teaching assistants were available to help students, and the “lecture” was relatively short and focused. Thus, we learned that students do not necessarily want to proceed in an unguided fashion: the feedback regarding this extra lab/lecture was very positive. It is often assumed, however, that all students enjoy working independently with little supervision. It is notable that

this lecture was particularly popular with non-traditional students and underrepresented minorities and women.

Information Technology at the Campus Level

Several of the changes we made when we proceeded with the redesign of our course greatly increased the complexity of the “information technology” (IT) involved. In utilizing on-line materials, we needed to have some of them installed in public lab sites so that they would be accessible to students. This in turn involved IT personnel from the central campus computing facility, who needed to know a semester ahead of time what software would be used. It was not always possible to provide this amount of lead time: new software is released very frequently, and publishers may not have new versions of software available until right before (or sometimes after) the start of a semester.

The CMS was administered by a separate group of IT professionals who also were in charge of installing software in lecture classroom computers. In terms of computer hardware and software, the computers in the lecture classrooms (where we often demonstrated software to students) were different from those in the labs where students did their work. The desktops looked different, and could not be changed by faculty or students as is customary in a shared computing facility. However, such differences are a concern, particularly with beginning students who are easily confused. These two sets of computers were administered by different IT groups, so glitches in the lecture classrooms were unrelated to those in the labs and had to be handled separately. Faculty have the option of using their laptops in the lecture classrooms, however, again, they do not “look” like the students used by the students.

The LAN-based testing software we utilized required installation of client software in public computing sites that communicated with a server in our department. Thus, the testing required the services of the systems administrators in our department for the server, the IT personnel from central campus computing for the public site client installation, and an additional IT person from a separate IT facility for the Engineering School to assist with networking problems. Troubleshooting the LAN-based testing software required not only communicating with three different IT groups, but also frequent email and telephone communications with the publisher. The system ran so slowly that it was virtually unusable at times. It took some students 45 minutes before their scores were received by the server after taking an exam. Unfortunately, we were not able to determine precisely what caused the problems due to the number of possible contributing factors. It could have been the network in the computer lab or the network at any point between the lab and our department. It could have been a hardware problem with the clients or server. It could have been a glitch in the publisher software. With so many different machines, software, people, and network connections involved, tracking down problems is extremely time consuming and may not produce results. In particular, publisher developed software is not well-understood by the publishers. It is generally produced by a contractor who may not have provided the publisher with extensive documentation.

Cost Savings: Time and Money

There may be opportunities for cost savings when on-line technology is utilized in a course. One of the goals of the Pew Learning and Technology Program is to demonstrate cost savings when electronic resources are incorporated into teaching. We anticipated a reduction in faculty time spent on the course, both in terms of administration and in developing materials. This, obviously, was not realized. We did realize a cost savings from some of the other changes made to the course, including the use of undergraduate rather than graduate teaching assistants, and the shift of the labs from a departmental to a university-maintained facility. Substituting undergraduate assistants for graduates was made possible by the use of on-line learning materials. The role of the assistants in the lab changed from lecturer to helper, where most of the demands on the assistants involved helping students with technology.

Problems: The Moving Target

There are two primary causes for the problems we encountered. First, the technologies are still relatively new. We expect CMSs to improve and evolve in response to user needs and input. The infrastructure for IT may not be as advanced as is required for technology-intensive uses. We found that every semester there was a major change with which we had to grapple, including new versions of operating systems, new versions of the applications software, new versions of the testing software, updated teaching materials, new formats for grade reporting, new hardware, etc.: each such change required reinstallation (sometimes with disastrous results), upgrading, or relearning on the part of the instructor and the IT personnel.

Second, our subject matter, computer science, is itself a “moving target”, with considerable updating of lecture and teaching materials required on a regular basis. Thus, unlike some courses or disciplines where a relatively static course might be designed and put into a CMS, the course content in our case changes often enough that the effort required to develop our own on-line materials is not worthwhile. It is easier to alter lectures than to rewrite an on-line course. Publishers provide CMS content for some of their books, but we have found that relying entirely on publisher-produced materials is not sufficient.

With technology continually changing as well as our subject material, we were faced with particularly difficult challenges.

Conclusions

As this anecdotal report demonstrates, incorporating technology into teaching is not always easy. What is fun and exciting for students is not always best for faculty: while the faculty for our course were burdened by many of the difficulties encountered, students responded positively to the changes^{3,4}. On-line teaching materials were well-liked. Students enjoyed longer laboratory sessions. Undergraduate assistants were preferred over graduates, in part due to the students' perception of them as helpful peers.

One of the promises of on-line technologies has been the “ease” of teaching more students with fewer faculty. Our experience has been that it is difficult to manage large classes, even with the help of on-line technologies. The best experiences with on-line teaching seem to come from small courses where students and instructors communicate with one another frequently. For

some disciplines it may be easier to develop on-line materials, but still, it must be asked wherein the satisfaction lies when one faculty member is “teaching” over a thousand students.

The currently developing OKI project at MIT⁵ represents a positive step towards creating open-source software for course management that can be used to integrate educational resources from a variety of sources. The project involves faculty from different institutions collaborating to help to solve some of the problems we have identified. We believe that good solutions will come from faculty who understand computing practices and problems in addition to having experience teaching with electronic course materials.

In summary, this chronicle of our experiences raises issues that many faculty who use a CMS may encounter. It is our hope that this faculty view of the IT revolution might be useful to other faculty as well as administrators, publishers, and software vendors.

1. Pew Learning and Technology Program, <http://www.center.rpi.edu/pewhome.html>
2. Virginia Tech Math Emporium, http://www.emporium.vt.edu/index_main.html
3. D. Walters, D. Burhans, H. Kershner and C. Alphonc, "Using Technology to Increase learning and Decrease Costs: A Collaboration with the Pew Learning and Technology Program., CIT 2000: Conference on Instructional Technologies, Buffalo, NY, 2000, <http://pew.cse.buffalo.edu/publications.html>
4. D. Walters, H. Kershner, C. Alphonc, D. Burhans, "Using Technology as a Change Lever: Increasing Learning and Decreasing Costs in a Computer Fluency Course", *EDUCAUSE 2000*, Nashville, TN, October, 2000.
5. MIT OKI Project, <http://web.mit.edu/oki/>

DEBRA T. BURHANS

Debra Burhans is an Assistant Professor in the Department of Computer Science at Canisius College in Buffalo, where she also serves as the director for the Bioinformatics Program. Her research interests include computer science education, bioinformatics, and question answering.

CARL ALPHONCE

Carl Alphonc is a Teaching Assistant Professor in the Department of Computer Science and Engineering at the University at Buffalo. His research interests include natural language understanding, programming languages, and computer science education.

HELENE KERSHNER

Helene Kershner is Assistant Chair and a Lecturer in the Department of Computer Science and Engineering at the University at Buffalo. She is the author of a book on computer literacy and has research interests in computer science education.

BARBARA SHERMAN

Barbara Sherman is a Teaching Assistant Professor in the Department of Computer Science and Engineering at the University at Buffalo. She has research interests in statistical analysis of data, management information systems, and computer science education.

DEBORAH WALTERS

Deborah Walters is Associate Dean for Undergraduate Education in the College of Arts and Sciences at the University at Buffalo where she is also an Associate Professor of Computer Science and Engineering. Her research includes work on learning styles in virtual environments, the development of web-based tools to support collaborative learning, and the development of woman-friendly environments for learning information technology.