British Journal of Educational Technology Vol 34 No 5 2003 639–650

# Engineering an online course: applying the 'secrets' of computer programming to course development

**Timothy J. Ellis and William Hafner**

*Timothy J. Ellis, PhD, teaches courses in Multimedia Systems, Educational Database Systems and Online Learning Environments at both the MS and PhD levels. Areas of research interest include multimedia, application of database technology, distance education and adult education. Address for correspondence: Dr. Timothy J. Ellis, Graduate School of Computer and Information Sciences, Nova Southeastern University, Fort Lauderdale, FL 33314 USA; web site: http://scis.nova.edu/~ellist; email: ellist@nova.edu. William Hafner, PhD, teaches courses in Knowledge Management, Information Retrieval and Software Development at both the MS and PhD levels. Areas of research interest include knowledge management, information retrieval, security, and software development. Address for correspondence: Dr. William Hafner, Graduate School of Computer and Information Sciences, Nova Southeastern University, Fort Lauderdale, FL 33314 USA; web site: http://scis.nova.edu/~hafnerw; email: hafnerw@nova.edu.*

**Abstract**

Colleges and universities are increasingly migrating towards utilising the World Wide Web to convey at least part of, and in many cases, their entire curricular offering. Despite this trend there is little support for the professors responsible for translating courses refined over a career in the classroom for delivery via the Web. Teachers who are experts in their subject area and masters of their craft when in a classroom find themselves in the uncomfortable position of having to relearn how to teach in a new environment with little or no support. Development of an online course is, in many significant aspects, analogous to developing a computer product. The procedures and tools utilised in the software engineering field to manage computer software development, therefore, offer promise for developing online courses. This paper explores the potential of one process developed for the software engineering field—the System Development Lifecycle (SDL)—as a tool to effectively design and develop online college courses.

## Problem statement

The allure of online courses is quite appealing to both academic institutions and their students. Colleges and universities are increasingly migrating towards utilising the World Wide Web to convey at least part of and, in many cases, their entire curricular offering (Mayor, 2001). Despite this trend there is little support for the professors responsible for translating courses refined over a career in the classroom for delivery via the Web (Muilenburg and Berge, 2001). Teachers who are experts in their subject


Published by Blackwell Publishing Ltd., 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

area and masters of their craft when in a classroom find themselves in the uncomfortable position of having to relearn how to teach in a new environment with little or no support.

Although there is no shortage of products such as WebCT and Blackboard available for distributing educational materials in an online modality, there is a definite lack of products that help the instructor plan, design and develop online educational programs. Despite the rather free availability of general suggestions and guidelines (Carlson and Everett, 2000; Levin *et al*, 1999), the professor is often left to develop her or his own answers to three important questions: What online resources are available? Which resources are appropriate for a given educational goal? How can those resources be effectively utilised to attain that goal?

Teaching, regardless of the environment, is neither easy nor assured of success. Irrespective of a teacher's efforts to help a student master a subject, the activity of learning is indeed one that can only be accomplished by the student. Since students differ widely in how they learn and what is effective in facilitating that process (Currie, 1995), designing an effective course is indeed challenging. The difficulty is exacerbated at the post-secondary level where teachers are primarily experts in the subject matter, not in educational processes and learning theory.

The nature of a web-based learning environment compounds the problem. Teachers, by-and large, teach in the same manner in which they were taught. Since few current teachers have experienced online learning as students, most are confronted with working in an environment for which they have no model. Furthermore the non-verbal cues that experienced teachers use to pace instruction in the classroom are not available online; the Web simply does not convey when a student's eyes glaze over. The immediacy of the question–response–follow-up cycle is largely lost in the online environment. It is quite difficult to foster and maintain the peer-to-peer support system of a community of learners when that community is 'virtual'. Many of the techniques—study groups and mentorship, for example—used in the classroom environment to help students master higher-level learning outcomes are difficult to replicate online. Thus, teaching online requires a re-evaluation of the pedagogical approach (Carr-Chellman and Duchastel, 2000; Dringus, 2000).

## An obvious solution: place class materials on the Web

The typical solution for implementing an online course is a simple porting process of placing materials developed for classroom delivery of the course on a web site supported by an online course management system such as WebCT or Blackboard. This process usually entails developing web pages containing class notes or copies of PowerPoint presentations, video-taped lectures, and various synchronous (ie, chat rooms) and asynchronous (ie, threaded discussion forums) simulations of classroom discussion (Carlson and Everett, 2000). Although the course management system provides structure for the course materials, there is no assurance that the materials that were effective in a classroom will be equally effective online.

While one cannot help but use prior experiences in teaching a course when translating that course for delivery via a new medium, the direct porting of classroom materials to the Web is usually not effective. Often the porting of the materials for web delivery is left to a support technician who lacks expertise in both the subject matter and pedagogical theory. When the instructor or professor does the porting of the class materials to the Web, a lack of experience with the tools available in an online learning environment and a general lack of technical expertise can serve as significant impediments (Muilenburg and Berge, 2001). Regardless of how the porting of the materials is accomplished the critical consideration is often ignored: the online learning environment is essentially different from the classroom environment and the techniques that worked quite adequately in the classroom might well be totally ineffective in a web-based course (Carr-Chellman and Duchastel, 2000; Dringus, 2000).

The problems with directly applying educational expertise developed in the classroom to the web-based environment do not end with course delivery. Meaningful evaluation of the effectiveness of the course is essential (Angelo and Cross, 1993). Often, however, online courses are evaluated using generic tools developed for the classroom environment. The value of these tools in evaluating and improving classroom instruction has long been questioned. Their value, when applied in the significantly different online environment, is even more questionable since they have never been validated for that environment and response rates are historically quite low. Meaningful evaluation that could be used to monitor and improve online learning experiences is, at best, difficult (Izat *et al*; Ravelli, 2000).

**Learning from software engineering**

It is clearly not enough to present even an experienced classroom-based instructor with a set of tools for teaching online. Development of a course for delivery via an online environment is in many ways similar to developing a computer product. Development of a computer product such as a game or multimedia-enriched computerised encyclopaedia relies upon the work of a team consisting of many different members—project manager, designers, content developers, programmers and subject matter experts—following a structured process model. It is unreasonable to expect a college professor with no development experience, minimal team support and arcane development tools to create an effective web-based educational product.

Since development of an online course is, in many significant aspects, analogous to developing a computer product, the procedures and tools used in software engineering would appear to be appropriate for developing online courses. The System Development Lifecycle (SDL) is the commonly accepted method of addressing the need for structured teamwork (Pressman, 1997).

The SDL views the development of a product as a sequential, iterative process. The process is sequential in that it is comprised of a series of distinct steps that must be accomplished in order, and is iterative in that issues that arise at any given step can necessitate a return to any previous step for reassessment.

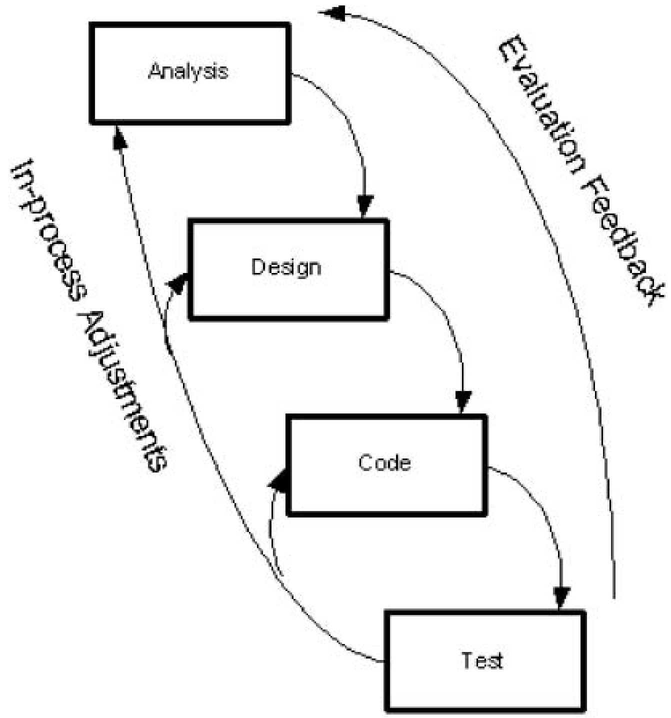



*Figure 1: System development lifecycle*

There are many different process models for SDL—waterfall or linear, spiral, incremental and concurrent, to mention just a few—and many different views of the number and names for the steps in the development process. Pressman (1997) offers perhaps the seminal analysis of the various models used in the development of software systems. For the purposes of this discussion, a rather simplified adaptation of the waterfall with feedback model seems most appropriate (Figure 1). This model is based upon a linear design with four major steps: analysis, design, code and test.

The analysis step entails answering four questions: What are the goals for the program? How is the program supposed to attain those goals? What data must the program have in order to attain those goals? How should the user be able to interact with the program? A requirements specification that details the desired functionality and sketches a general model for the desired product is typically developed to answer these questions.

In the design step the model sketched during analysis is refined. Detailed descriptions of the product architecture, user interface and necessary software components are developed. The design process entails translating the rather general requirements identified in analysis into a highly detailed blueprint for the program.

During the coding step the blueprint developed during program design is implemented. Coding entails writing the machine-executable instructions necessary to build the program that can meet the requirements specified during the analysis phase.

During the test step the program is subjected to a series of planned, structured trials to determine if it meets the requirements detailed during the analysis step. There are generally two aspects to testing: verification to ensure all functions perform as designed, and validation to ensure that the product does in fact meet the goals specified in the requirements specification.

A key element is the feedback loop from the test step back to analysis at the start of the process. This process, when applied over time, is intended to produce continuous improvement in the product based on the test results. As well, at each step in the process there can be a return to a previous step to clarify or revise the results of that step in the process. In fact, as depicted in Figure 1, when a problem is discovered while working on developing the concepts in a specific step, any previous step in the process could be revisited if necessary to resolve the problem.

## Course development lifecycle (CDL)

The processes and controls inherent in the SDL have applicability to course development, especially for those courses developed for delivery in an online format. The following sections examine each of the steps in the SDL and the corresponding step in a CDL approach to course development. Figure 2 illustrates the CDL. The discussion is illustrated by examples from a course in Educational Database Systems that was developed for online delivery using the CDL process model.

### *Analysis by developing learning outcomes*

Learning outcomes form the essence of a requirements specification for a college course (Bloom *et al*, 1956; Gronlund, 2000). The learning outcome must clearly specify what the student should be able to do, the conditions under which the student should produce the desired behaviour and how well the student must be able to perform it. In addition to focusing on the specific behaviour, learning outcomes must also reflect the level of cognitive activity expected of the student.

Although educational researchers have developed many different taxonomies for categorising level of cognitive activity, the work presented by Bloom *et al* (1956) is an effective model for developing learning outcomes. Bloom identified a sequence of six levels of cognitive activity and associated specific types of student behaviour for each. Table 1 summarises those levels and a representative sample of associated behaviours.

The process of developing learning outcomes entails three steps: (1) identify the specific topic or subject matter to be learned, (2) decide what level of cognitive activity is necessary for satisfactory mastery of that topic and (3) identify what student behaviour is

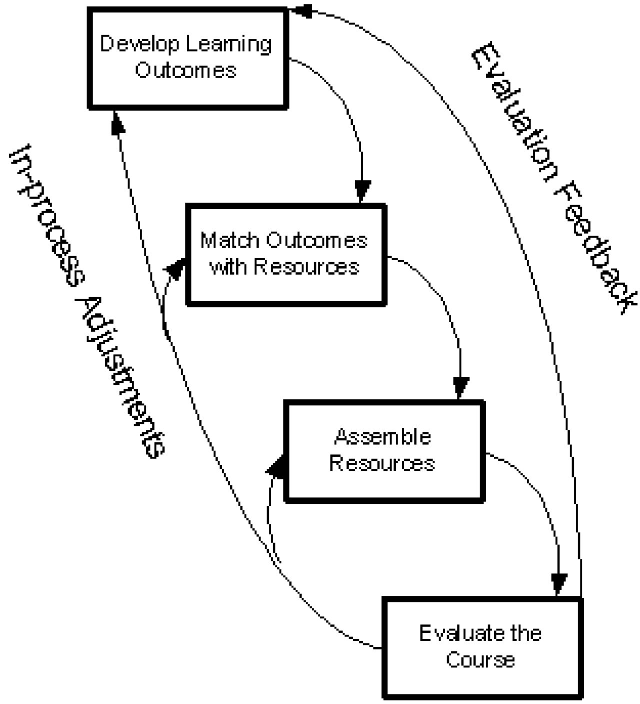



*Figure 2: Course development lifecycle*

*Table 1: Bloom's levels of cognitive activity and associated behaviours*

| *Cognitive level* | *Samples of associated student behaviour* |
|---|---|
| Knowledge | list, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name |
| Comprehension | summarise, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend |
| Application | Demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover |
| Analysis | separate, order, explain, connect, classify, arrange, divide, compare, select, explain |
| Synthesis | combine, integrate, modify, rearrange, substitute, plan, design, invent, compose, formulate, prepare, generalise, rewrite |
| Evaluation | assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare |

1. Plan, develop, and document an educational database management system.
2. Work effectively as a member of an educational database development team.
   a. Collaboratively develop a requirements document
   b. Collaboratively develop a production schedule
   c. Collaboratively develop an entity-relationship diagram
   d. Collaboratively develop a data dictionary
   e. Collectively produce an educational database application such as a computer managed instruction (CMI) or computer assisted instruction (CAI) program.
3. Identify and analyze the technological impediments to implementing computerized database management system solutions in an education setting.
4. Analyze educational database implementations, identifying strengths and weaknesses based upon a review of appropriate literature.
5. Evaluate problem areas in the education field and analyze the appropriateness of a computerized database management system as a solution.
6. Produce meaningful original work in the area of applying computerized database management systems in educational settings.

*Figure 3: Sample learning outcomes for educational database course*

necessary to demonstrate that cognitive activity. For example, in an introductory database course, normalisation is a core topic area, students are commonly expected to progress to at least the application level of cognition, and typically students demonstrate that mastery by solving one or more problems. The learning outcome describing this expectation might be: The student will be able to solve a database normalisation problem by bringing a dataset into Third Normal Form compliance. Figure 3 presents a sample of the learning outcomes developed for the Educational Database Systems course.

*Design by matching outcomes with online resources*

Once the learning outcomes have been identified for a course, the instructor must design the course by determining which pedagogical tools should be used to facilitate the students' attainment of each specific outcome. There are five components to the design phase. First, a decision must be made regarding the time-dependency of the course: asynchronous, synchronous or hybrid. Second, a catalogue must be developed that lists the pedagogical tools available for the selected time-dependency, the effective uses for each tool and the known limitations of each. This catalogue must be based upon local experience; differing student profiles, faculty profiles, curricula and even courses require different pedagogical approaches. Figure 4 presents a partial list of the catalogue of online pedagogical tools available for the Educational Database Systems course developed for asynchronous-only delivery.

After the resource catalogue has been developed, the instructor can then select the appropriate instrument or instruments for each learning outcome. Merely identifying a tool or set of tools to facilitate attainment of a given learning outcome is, of course,

| **Pedagogical Tool** | **Features, Uses, and Limitations** |
|---|---|
| Email | • One-to-one and one-to-many communication<br>• Teacher-student(s), student-teacher, student-student(s)<br>• Primarily text, but graphic and audio potential<br>• No or minimal pre-course preparation<br>• Moderate to heavy in-course attention<br>• Minimal demands on student's system |
| Threaded discussion board | • One-to-many communication<br>• Teacher-students, student-teacher, student-students<br>• Supports student-student collaboration<br>• Primarily text, but full multimedia capability through attachments<br>• Minimal pre-course preparation<br>• Moderate in-course attention<br>• Minimal demands on student's system |
| Online PowerPoint presentation | • Convey course content normally delivered by demonstration.<br>• Full multimedia potential (narration, animation, video, etc.)<br>• Moderate to heavy pre-course preparation<br>• Minimal in-course attention<br>• Moderate demands on student's system |
| Streaming video | • Convey course content normally delivered by lecture.<br>• Full multimedia potential (narration, animation, video, etc.)<br>• Heavy pre-course preparation<br>• Minimal in-course attention<br>• Heavy demands on student's system |
| Group practical application, completed locally on the computer, but developed collaboratively through use of email and threaded discussion forum | • Provide avenue for constructive learning environment through collaborative, hands-on learning opportunities.<br>• Hybrid stand-alone and interactive experience<br>• Moderate pre-course preparation<br>• Moderate in-course attention<br>• Moderate demands on student's system |
| Peer-reviewed research paper | • Encourage and measure acquisition of the learning outcomes that are process oriented.<br>• Student-to-teacher and student-to-student interactivity<br>• Moderate pre-course preparation<br>• Heavy in-course attention<br>• Minimal demands on student's system |
| Technical briefs | • Encourage and measure acquisition of learning outcomes that are content oriented<br>• Student-teacher, teacher-student interaction<br>• Moderate to heavy pre-course preparation<br>• Moderate to heave in-course attention<br>• Minimal demands on student's system |

*Figure 4: Catalog of online tools*

1. There are four (4) discussion topics. Each topic is worth ten (10) points for a total of 40 points or 20% of your course grade.
2. All contributions must be made in the SCIS Student Forums page.
3. All contributions must be made during the specified time period for the topic. Contributions made before or after the specified period will not be accepted unless previously approved by the instructor.
4. Up to five (5) points will be awarded for contributions as follows:
    - Strength of insight of the contribution
    - Support from the literature (2 solid citations)
    - Interest sparked, as demonstrated by quality responses by others
    - You may make as many entries to a discussion topic area as desired during the period in which that topic area is open for contributions.
5. As a graduate student you are expected to be proficient in the use of the English language. Errors in grammar, spelling, or syntax will affect your grade. As your professor, I will not provide remedial help for writing problems. If you are unable to write clearly and correctly, I urge you to contact the program office for sources of remedial help.
6. Discussion topics
    1. One of the most obvious uses of database management systems in education is the electronic library system. There are several different electronic library databases available. Which is your favorite, and why? Be sure to discuss its strengths and weaknesses.
    2. Search engines can only be as effective as the search criteria specified by the user. What makes a search effective and what can you do in phrasing your search string to optimize the search?
    3. Discuss a particularly effective implementation of a database management system in an educational setting.
    4. Discuss an area of particular need in education, which you think can be effectively addressed with a computerized database management system.

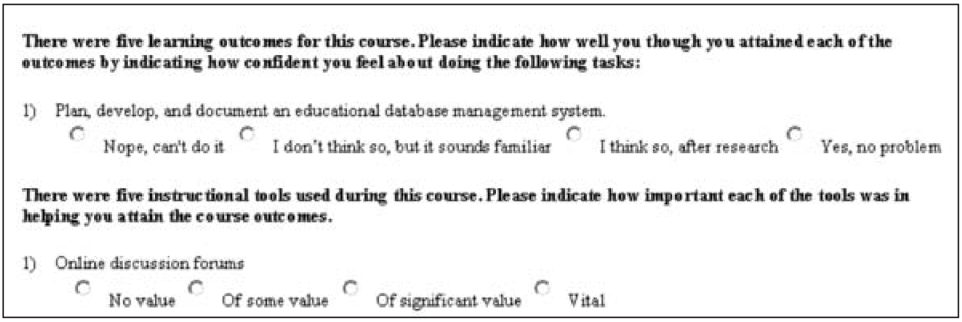
*Figure 5: Sample assignment using threaded discussion forum*

not adequate; the instructor must detail how the tool will be implemented in terms of assignment specifics. Finally, the set of assignments supported by the online pedagogical tools must be integrated into a coordinated whole that is comprehensible and usable by the student. This integration can be accomplished either through an interactive syllabus page in which each assignment is detailed and contains a link to the appropriate online tool or tools, or through a coordinated course management system such as WebCT. Figure 5 details how one of those tools—the threaded discussion board—was implemented in the Educational Database Systems course.

*Code by assembling the resources*

During the code phase the course as designed is built or customised. Videos are filmed, compressed and placed on web sites; threaded discussion forums are created; chat rooms are established; class notes are converted to web-deliverable format; and the entire package is assembled into a seamless whole ready for student use. This stage is the point in the development process where most online educational tools such as Learning Space, WebCT and Blackboard start.

In software development it is a normal practice to use a team approach. The skills necessary for effective product design are usually quite different than the coding skill set. It is not unusual for those responsible for product design to delegate the actual coding to other members of the team. In a similar fashion it is unrealistic to expect the instructor to build and customise the various elements necessary for the online delivery of a course. Just as one would not expect a professor to write the textbook and build the classroom used to support a course delivered in a traditional, face-to-face environment, one should not expect that individual to fulfil all the roles inherent in the course development process.



**There were five learning outcomes for this course. Please indicate how well you though you attained each of the outcomes by indicating how confident you feel about doing the following tasks:**

1) Plan, develop, and document an educational database management system.
Nope, can't do it · I don't think so, but it sounds familiar · I think so, after research · Yes, no problem

**There were five instructional tools used during this course. Please indicate how important each of the tools was in helping you attain the course outcomes.**

1) Online discussion forums
No value · Of some value · Of significant value · Vital

*Figure 6: Sample course evaluation questions*

*Test by delivering and evaluating the course*

There are two essential questions that must be answered by a course evaluation: have the learning outcomes identified for the course been attained (course effectiveness), and did the pedagogical tools utilised in fact facilitate the attainment of those outcomes (course efficiency)? The evaluation must be an integral part of the course delivery in that it completes the development lifecycle. Evaluation serves as a vital part of continuous course improvement by providing input to the analysis for the next iteration of the course. The results of an effective evaluation provide the data necessary for the evaluation feedback loop in the CDL depicted in Figure 2.

Typically online courses are evaluated with the same general-purpose instrument used to evaluate classroom-based courses. This type of evaluation suffers from two obvious flaws: it does not adequately accommodate the unique characteristics of the online course, and, perhaps more importantly, it is too general in nature to offer meaningful insight into the effectiveness and efficiency of the course regardless of delivery modality (Stanfel, 1998). It is hard to imagine what is truly being measured by evaluations comprised of questions such as 'How would you rate the course?', 'Was the instructor approachable?', 'Rate the course organisation' and 'Rate your confidence in the instructor's knowledge'. An evaluation customised for a specific course that asks the student to first rate her or his level of confidence regarding attainment of each of the learning outcomes specified for the course and then rate the value of each of the pedagogical tools incorporated in the course would at least appear to offer both content and criterion validity. Assessment of the course should be "learner-centered, teacher-directed, mutually beneficial, formative, context-specific, ongoing, and firmly rooted in good practice" (Angelo and Cross, 1993, 4). Figure 6 illustrates an initial attempt to address meaningful course assessment from the evaluation of the Educational Database Systems course.

## Summary and implications for further research

Developing an effective college-level course is indeed complex and challenging. When the need to adapt the course to an emerging, poorly defined and imperfectly understood delivery system such as an online learning environment is added to the developmental burden, the difficulty becomes exponentially greater. The structure and support inherent in a formalised development process offers great promise as a means of addressing this difficulty.

Without question, courses have been successfully planned, developed and delivered via online modalities. There are certainly a large number of process models that present potential as a solution to the problem of developing courses for an online environment. This paper explored the potential of adapting the SDL model widely used in software engineering as a particularly promising process model. There is an intuitive appeal to adapting a tool developed by the computer industry to plan a course that will be delivered predominately via computer-mediated communication.

The CDL process model presented in this paper is by no means the ending point of this discussion. One of the greatest strengths inherent in the structure of the SDL model is the capacity to support the development process through computerisation with Computer Aided Software Engineering (CASE) tools. This paper represents a start of the analysis phase for a computerised support system for course development. Future research identifying specific requirements and design considerations for such a system is indicated.

## References

Angelo T A and Cross K P (1993) Classroom assessment techniques: a handbook for college teachers (2nd ed.) Jossey-Bass, San Francisco.

Bloom B S, Engelhart M D, Furst E J, Hill W H and Krathwohl D R (1956) (eds) Taxonomy of educational objectives, the classification of educational goals, Handbook I: cognitive domain Longman, New York.

Carlson R and Everett D R (2000) Taking instruction online? Paper presented at Mid-South Instructional Technology Conference, Murfreesboro, TN.

Carr-Chellman A and Duchastel P (2000) The ideal online course British Journal of Educational Technology **31**, 3, 229–241.

Currie G (1995) Learning theory and the design of training in a health authority Health Manpower Management **21**, 2, 13–19.

Dringus L P (2000) Towards active online learning: a dramatic shift in perspective for learners The Internet and Higher Education **2**, 4, 189–195.

Gronlund N E (2000) How to write and use instructional objectives Prentice Hall, Upper Saddle River, NJ.

Izat J G, Mc Kinzie L, Mise C D and Mcallie T (2000) Evaluating of world wide web delivered university courseware: creating an instrument appropriate for a new course delivery medium in Society for Informational Technology & Teacher Educational Conference: Proceedings of SITE 2000, 8–12 Feb, San Diego, CA.

Levin J, Levin S R and Waddoups G (1999) Multiplicity in learning and teaching: a framework for developing innovative online education Journal of Research on Computing in Education **32**, 2, 256–269.

Mayor T (2001) E-Learning: does it make the grade? CIO Magazine **14**, 7, 132–144.

Muilenburg L Y and Berge Z L (2001) Barriers to distance education: a factor-analytic study The American Journal of Distance Education **15**, 2, 7–22.

Pressman R S (1997) Software engineering: a Practitioner's Approach ($4^{th}$ ed.) McGraw-Hill, New York.
Ravelli B (2000) Anonymous online teaching assessments: preliminary findings. Paper presented at AAHE Assessment Conference 2000: Rising Expectations: Can Assessment Deliver? 14–18 June, Charlotte, NC.
Stanfel L E (1998) Accuracy of teaching evaluation information systems: a case study with analysis and re-design Journal of Instructional Psychology **25**, 2, 82–93.