

## **WEB CONFERENCING SOFTWARE IN UNIVERSITY-LEVEL, E-LEARNING-BASED, TECHNICAL COURSES**

**JOHN W. COFFEY**

*The University of West Florida, Pensacola*

### **ABSTRACT**

Faculty who offer university-level technical courses through e-Learning environments must surmount several difficult challenges. Traditionally, learning in technical courses takes place through real-time interactions between instructor and students in the course of solving actual problems—a form of activity that is not possible in asynchronous e-Learning systems. The current study contains an account of experiences gained in using Web Conferencing software to provide a spoken synchronous component to a project-oriented senior-level computer science class. Strengths and weaknesses of this undertaking and a sketch of a different type of hybrid course are presented. While the discourse in this article is focused on computer science, principles put forth are applicable to teaching and learning in any technical field.

### **INTRODUCTION**

Delivery of computer science and other technical courses has traditionally involved face-to-face teaching and mentoring, with a significant in-class component, and active learning projects in the form of programming assignments, the development of software and database designs, discrete mathematics problem solving, etc., outside class. Lecture and discussion-based in-class activities generally play a salutary role in helping students surmount the many technical

challenges inherent in such courses. However, many faculty nationwide (including those in computer science) are tasked with developing on-line courses [1]. Consequently, a rethinking of pedagogical approaches to the teaching and learning of technical curricula in order to exploit the strengths and ameliorate the deficiencies in the online environment is due.

Online course delivery raises particular issues and challenges in technical fields. Challenges include how to compensate for the role of interactive teaching and guidance in the mastery of complex problem solving such as computer programming and mathematics, how genuine mentoring can be achieved, and the quality of learning that takes place and its assessment. Employing Web Conferencing software such as Elluminate or Connect Pro [2, 3] holds potential to provide at least a partial answer to some of the more problematic aspects of teaching/learning technical material online. However, relatively little has been written regarding best practices for the use of this technology, particularly within technical disciplines, and from the point of view of experiences with actual class offerings.

The remainder of this article will address relevant literature pertaining to methods and outcomes in online computer science courses and methods, benefits, and drawbacks of online learning activities that afford spoken communication and sharing of electronic resources. This article contains a description of a model for blended courses in which the face-to-face component of a hybrid course is handled with Web conferencing software. It also contains a case study in the use of Web Conferencing software to support a variety of interactive online activities in an upper level computer science course. This work closes with a summary and conclusions regarding lessons learned.

## BACKGROUND

This section contains a review of background literature relevant to online teaching and learning in computer science courses. Although the literature pertaining to online course delivery is extensive, it does not offer a great deal of guidance regarding how to exploit new technologies to deal with those problems particular to courses such as computer programming, nor how to perform valid assessment in technical courses for which a great deal of relevant information is obtainable online. The literature provides little at all addressing the use of Web Conferencing activities in this particular online setting. The following sections contain descriptions of literature pertaining to methods and assessment in online computer science courses.

### The Conduct of Online Computer Science Courses

Thomas [4] described an early offering of an online C++ programming course. She related that even though the course was geared toward mature students with at

least a year of programming experience, face-to-face meetings with a teaching assistant proved useful. Several students reported feeling isolated and wishing for better contact with the instructor. Thomas offered a variety of suggestions including holding face-to-face reviews and exams. Thomas concluded that online programming courses require mature and motivated students and that ways to compensate for the lack of face-to-face interactions would be helpful.

Molstad [5] developed an online version of a Business Applications Programming course. She stated that students brought a range of capabilities to the class, and that the better-prepared students appreciated not having to sit in class while concepts they already understood were discussed. She stated that students with no prior programming experience had some trouble with the assignments and labs. She also described the use of two-way audio-video capabilities that were used to allow students to access recordings of lectures that pertained to the course.

Zachery and Jensen [6] describe the organization of a course in Javascript programming that is offered online. Their course provided example-based narratives (focused descriptions of individual concepts), exercises, a finished version of the case study application that students were asked to build, and a series of assignments. On the basis of 3 years of offering the course, they deem their approach to be a success. The authors further noted that students could get help via telephone calls to faculty during office hours, maintaining a spoken interaction mode in the course. While they correctly relate that online courses should not simply attempt to replicate the face-to-face experience, it is true that a lack of face-to-face experience has differing impact upon different types of courses.

For instance, Hirshheim [7] described a variety of important capabilities that are lost when conducting technical courses online, including the loss of capability to work problems interactively or to hold interactive discussions, the loss of visual cues regarding student comprehension, and the loss of ability to enforce individual work. While advocating for purely online computer science courses for students who would not otherwise have access to such curricula, Gal-Ezer [8] bemoans the lack of formal instruction and basic guidance available to teachers of such courses. She also states that blended mode courses with a face-to-face and online component potentially offer the best attributes of both of these course delivery methods.

Attempts have been made to individualize technical course instruction online through the use of intelligent tutoring systems [9, 10]. Interestingly, while stating that the apprenticeship model with its very high ratio of journeymen to apprentices might be the most effective way to teach programming, Shaffer [11] describes an online tutoring system to help students learn to program. The system, named Ludwig, is an attempt to provide students with the individualized attention provided by a master-apprentice relationship and potentially lacking in an online setting.

Gill [12], an advocate of online education, describes a range of technologies for online teaching of IS courses, including the use of Elluminate, an interactive,

Web Conferencing tool, citing uses for group problem solving and code sharing. He states that online teaching significantly changes the nature of the teaching task with interactive online course activities being less effective than similar activities performed in a face-to-face environment, but asynchronous activities opening up new learning opportunities. Interestingly, he places emphasis on changes in student expectations regarding how they will be assessed as opposed to difficulties with assessment itself.

Coffey [13] describes benefits and drawbacks in the use of Web Conferencing for online computer science courses. He suggests that the main benefit lies in the increased information bandwidth compared to e-mails going back and forth for the resolution of technical problems. He also cites the opportunities for assessment based upon the interchange afforded by talking back and forth to add validity to the assessment process in online courses.

Web Conferencing activities might play a role in preparing students for future work environments. Kazmer and Haythornthwaite [14] suggest that they are concerned about how students will develop a sense of community when courses are increasingly offered online. Web Conferencing software might help to address this issue. Friedman [15] described a changing landscape of work in which collaborators might be located anywhere in the world. Providing collaborative learning activities via Web Conferencing sessions might afford preparations for the sort of work that future computer professionals are likely to perform.

A recurring theme in the literature reviewed so far is the need for efficient communication paths for students and instructors in e-Learning technical courses. Efficient communication is critical in light of the types of problems students confront. For instance, it is often the case that a minor, easily fixed problem might cause the student to stall until help arrives, and e-mail-based help might be some time in coming. Student motivation quickly suffers in such circumstances. Another important issue is the overall level of attainment and its assessment, the topic of the next section.

### **Outcomes and Assessment in Online Computer Science Courses**

This section contains a review of quality of outcomes that are achieved in technical courses that are presented online versus face-to-face. Ury [16] stated that in an absolute sense, performance of online students was satisfactory, but that their aggregate final grades were significantly lower than those of students who took an equivalent class face-to-face. El-Sheikh et al. [17] reported relatively little difference in the outcomes of students in face-to-face and online Java courses, but noted that dropout rates were much higher in the online versions of the course.

In follow-up work, White, Coffey, and El-Sheikh [18] reported that the students who had online versions of the introductory course did as well in the next programming course in the series as the students who took the face-to-face

version. Of course, the percentage of students who had gone on to the intermediate course from the online introductory course was lower than the cohort from the face-to-face introductory class, and likely were better students on average.

Courte [19] also addressed the completion issue with a direct comparison of online and face-to-face students on a technical problem (Javascript programming) and a nontechnical computer-related problem. She concluded that relative performance of the two groups was about the same on the non-technical problem, but much worse among the online students on the technical problem. This study again suggested that differences in the educational task exist depending on the knowledge being taught and learned.

Kleinman and Entin [20] suggested that there were no significant differences in overall outcomes between online and face-to-face classes. However, they noted that, in aggregate, the online students were older than the face-to-face cohort, and the technology itself was responsible for significant early attrition. Attrition in the face-to-face class was more gradual and remained significantly lower. These authors also suggested that the need existed for more responsive means to assist students with problems, both with the technology used to present the course, and in the course content itself.

While the attrition rate is a significant concern in online courses, assessment of attainment is an equally pressing issue. In technical degrees, answers to questions, computer programs, etc., are either right or wrong, and easily copied or shared. With so much computer programming code available on the Web, no effective ability to monitor who is helping whom with an online test or with programming assignments and other deliverables, etc., it is genuinely difficult to know how much students have really attained. An implication of this fact is that assessment becomes a critical problem in online technical courses.

Nelson, Settle, Bhagyavati, Shaffer, Miles, and Watts [21] make many points about best practices for online courses including the need for proctored exams as a gold standard for what students have learned. However, proctored examinations do not resolve the plagiarism problem. Although some commercial software can search the Web and compare files [22], and other programs such as those described by Lancaster and Culwin [23] and Cooper, Coden, and Brown [24] can compare submitted programming assignments for similarity, effective assessment goes well beyond the detection of cheaters. It is important to note that the problem of plagiarism also exists in face-to-face classes.

A final study on the topic of outcomes is informative. Evidence exists that students adopt different strategies in programming courses that are offered online [25]. Ebrahimi found that with less direct instruction, students tend to think in more varied and idiosyncratic ways about the problems they must solve and they often produced correct but complicated, difficult-to-understand solutions. This contrasted with the face-to-face students who produced simpler, more direct solutions. These results clearly create concern for the types of problem solving strategies that online students learn, and has implications for how they will do as

problems they confront grow larger and require solutions that are as simple as possible, comprehensible, and modifiable.

This literature review reveals some of the significant issues relevant to the education of college-level students in technical subjects offered increasingly in online environments. The need for useful demonstrations of effective problem-solving strategies, for efficient communication to attempt to keep motivation high and to address consistently higher attrition rates, and for effective strategies to perform assessment of learning outcomes are all valid concerns. The next section addresses ways in which Web Conferencing activities can address some of these issues.

### **USES OF WEB CONFERENCING**

Web Conferencing sessions can be used for a variety of useful purposes in technical courses. Kazer and Haythornethwaite say that online teaching is at its heart, a problem of moving information across space and time, aggravated by the “lean environment of text-based communication” [14, p. 7]. It is evident to anyone who has spent an entire morning trying to help a student resolve a relatively simple series of programming problems via e-mail, that they are fundamentally correct.

Web Conferencing sessions can improve communication by affording an experience that is similar to an office visit without requiring students to travel to campus. Such sessions can be used to review student programming code and to communicate with teaching assistants. Online office hours are an effective way to answer student questions more efficiently than through e-mail. Likewise, student programs can be debugged through Web Conferencing software. A typical feature of such software is the capability to share applications or the entire desktop of a computer. Faculty can either observe as students walk through a program looking for errors, or they can actually take over the computer to browse the code and figure out what is wrong. After determining the problem, the faculty can provide probe questions to help the student understand what has gone wrong, as would be done face-to-face.

Assessment is a critical issue in online environments for at least two reasons: to understand the day in and day out progress students are making and, ultimately, to assign a grade for the course. In online courses, it is necessary to evaluate student progress very frequently to try to understand what students understand and what they do not. Web Conferencing sessions afford great opportunities to assess more efficiently and frequently, and therefore, to gain better insight into the progress the students are making.

Through Web Conferencing sessions, faculty can perform reality checks on student attainment by having them explain how their programs work, how they identified and resolved design issues, etc. However, even with the help of Web Conferencing, problems such as who actually did the work or took the test,

and lack of ability to know what resources students employed when completing an assignment or taking an exam, still exist.

## **BENEFITS AND DRAWBACKS OF ONLINE CONFERENCING IN TECHNICAL COURSES**

### **Benefits**

Substantial benefits accrue to both the instructor and student by having Web Conferencing sessions in technical courses. The first and most general is that such sessions afford the possibility of counseling and mentoring students in a fashion similar to face-to-face sessions. Faculty in computer science have expressed some cogent reasons why asynchronous online computer science course delivery is potentially less effective than face-to-face delivery [7]. As enumerated here, significant reasons exist for this phenomenon beyond a reluctance to change pedagogical approaches. Virtual office hours might actually constitute an improvement over traditional office hours as more courses are slated online, students spend less time on-campus, and attending actual office hours increasingly entails a special trip to campus.

The possibility of recording sessions for later review by students is another major benefit of this type of course delivery. Many students miss classes for reasons that range from unavoidable circumstances to laziness. In traditional classes, their major form of recourse is to borrow a fellow student's notes. With recordings that enable them to review highlights and questions and answers, students might actually have a better means of dealing with missed classes with this delivery scheme. One of the really compelling aspects of Web Conferencing communication is the very high bandwidth of discussion relative to that of instant messaging or discussion threads.

It is clear that in online course delivery settings, Web Conferencing sessions can foster student motivation by helping them to avoid some of the pitfalls associated with the struggle to master complex problem-solving, essentially by themselves. It is reasonable to expect that greater success levels and reduced levels of struggle in learning how to program will improve retention. For instance, in debugging programs, it is time consuming for faculty to receive a potentially complex program via e-mail, to extract it to a folder, add whatever additional files might be necessary to make it execute, and see what happens. It is much more time-efficient to have the student join a session, share the application, and get on with resolving the problem.

### **Drawbacks**

Perhaps the most significant drawback to the implementation of Web Conferencing sessions are the costs associated with it, both for software and for faculty



and student time. Such software is not inexpensive to license and maintain, and in difficult budget times, it may be viewed as a luxury rather than a necessity. Additionally, not all institutions have the support staff to implement its use. Faculty time and student schedules are other critical issues. Student support in online courses is time-consuming, at least in part because students take such courses so that they can study (and potentially encounter problems) anytime. Each of these issues is discussed in turn.

Offering a Web Conferencing component places significant demands on the time of faculty. The author of this article has offered a hybrid online course with Web Conferencing and asynchronous components. In addition to the significant amount of work keeping up with e-mails and discussion threads, the instructor spent approximately half the time that would have been spent in a face-to-face class talking with students online. It is not known how much the Web Conferencing sessions decreased utilization of e-mail and discussion threads (which it certainly did), but it was clear that the time requirement to provide this service to students was significant.

Implications for student schedules present another issue. The impetus behind enrollment in many online classes is to provide education anytime, anywhere, thus avoiding synchronous activities whether face-to-face or online. Currently, many schools do not have a category for synchronous online classes. Discussion is currently underway at the author's institution to debate the merits of online courses scheduled in a time slot and to create a new category for such a course. The percentage of the course contact hours allocated to such activities and a policy regarding when the interactive components would be scheduled are still under discussion. Clearly a change of culture will be necessary as part of any significant adoption of this type of course.

Another perceived problem with this means of communication is that the instructor cannot utilize non-verbal cues to student understanding. An involved instructor can learn a lot about how the students are dealing with class proceedings by carefully monitoring student body language. In a face-to-face class, it is easy to see if students are really understanding, clearly confused, or turning off to the proceedings. Even with WebCams, it is difficult to attain good visual cues regarding student comprehension, and cameras are frequently not used in Web Conferencing sessions. Still, for all these problems, Web Conferencing can clearly enhance online learning. The next section describes a type of hybrid course with the Web Conferencing part essentially fulfilling the role of the face-to-face component.

### **A DIFFERENT VIEW OF BLENDED COURSES**

Research has shown that a blended approach to course delivery, part face-to-face and part online, is well-liked by students [21]. The use of Web Conferencing affords the possibility of a type of blended or hybrid course in which the



in-class, face-to-face component of a traditional blended course takes place in a Web Conferencing environment, and the asynchronous online component essentially stays as it has been before. The next sections contain a general discussion of the benefits of traditional hybrid courses and of this new model.

### **Traditional Hybrid Courses**

The part-online, part-face-to-face arrangement of blended courses provides many benefits for both students and faculty in technical curricula. The face-to-face component of this type of course provides the opportunity for high bandwidth conversations. In this component of the class, the instructor might lay out groundwork pertaining to basic knowledge to be applied in a problem area. For example, a discussion of a data structure that will efficiently solve a programming problem, how it is implemented, its best uses, and how it contrasts with other data structures that might be used can be explored in an efficient manner.

Open discussions in face-to-face classes can effectively address problems many students are likely to encounter. These problems can be anticipated by the instructor and discussed in a manner that enables every student to benefit from the discussion. Even those students who do not have the particular problem frequently gain insight into alternative approaches to the solutions of problems. Additionally, the face-to-face component of a blended class is an ideal time for assessment activities. Face-to-face classes afford the best possible level of control over the testing process by enabling the instructor to ensure that students are doing their own work.

The asynchronous portion in the traditional blended class affords the opportunity for more active learning for the student and the chance to try ideas that have been explored during the in-class component. Since technical courses routinely require a significant amount of outside-class activity by students (writing computer programs, solving problem sets, etc.), such individual efforts are necessary as a matter of course.

Since computer availability has become more pervasive, the asynchronous component also allows students to spend less time commuting to class and more time actively working on their studies at home. Other benefits include the fact that some students are reluctant to talk in class, but are more willing to participate if they have time to formulate questions or responses that are submitted to discussion boards. For all these reasons, an online course presentation model that embodies both Web Conferencing and asynchronous elements is well suited to technical course delivery.

### **Combining Asynchronous and Web Conferencing Activities Online**

When a synchronous web conference is combined with an asynchronous component, students benefit from a highly interactive environment without having

to go to a meeting place. Place-bound students, one of the major constituencies for which fully online courses are developed, can realize most of the benefits of hybrid courses without leaving home. At the same time, this version of a hybrid course still affords the asynchronous component that enables students to work anytime, anywhere on the course.

Typically, the Web Conferencing software embodies the notion of a moderator who can select the next student to speak and who can preempt overly long-winded or digressing discussions. For efficiency, communications are often half-duplex, but this design issue does not generally present a problem. In Web Conferencing sessions, an instructor can compensate to some degree for the lack of ability to process non-verbal cues by frequently asking whether or not the proceedings are making sense and to trust that students will be forthright about indicating when they are not.

Assessment still has some problems in this model. The most significant problem is that the Web Conferencing sessions greatly improve opportunities for oral testing, but provide little additional benefit for written assessment. The advent of polling/voting devices affords an enhanced capability for assessment of comprehension in Web Conferencing sessions, but does not contribute as much to assessment for grading purposes. It is also not as effective for assessment of problem solving where the answer may be wrong because of only one minor mistake.

Scheduling also remains a problem with this new blended model. The anytime, anywhere aspect of online courses is a major driver of student participation. It is innately difficult to get a single time that is good for every student. Instructors can schedule more than one session, but having even two Web Conferencing sessions per week entails spending as much time in the synchronous sessions as would be spent synchronously in a fully face-to-face course. Recording all the Web Conferencing sessions and holding online office hours partially ameliorates the scheduling problem, but still allows inequities in the student's access to the instructor and overall experience.

### **Migration to this Style of Course Delivery**

Migration to this new model is simple for blended face-to-face/distance courses. It is simply a matter of scheduling the Web Conferencing activities in such a way that students can participate, ensuring that students have headsets to talk and listen, and ensuring that the software is available. Migration is relatively simple for a fully online course. The potential difficulties lie in the possibility that some students take online courses because of severe scheduling constraints. Having a Web Conferencing component is likely to improve outcomes for those who can participate, but it might exclude or place at a disadvantage those students who cannot readily accommodate the schedule constraints. This style of hybrid course is easier to develop than a full migration of a face-to-face course to a fully

online course. Furthermore, initial deficiencies in an online version of a course can be ameliorated with the Web Conferencing component.

### **A CASE STUDY**

A semester-long, online version of a senior-level class in Programming Languages was offered for the second time recently. The class is a survey of programming language paradigms and features with non-trivial programming assignments in a variety of programming languages and development environments. The course is a standard part of most computer science curricula. The following sections contain a case study that provides details regarding the conduct of the class and the role of Web Conference-based learning events in the class.

#### **Course Content, Students, and Technologies**

Broadly speaking, the content of this course includes basic issues pertaining to the design and implementation of programming languages. Goals for student learning range from detailed understanding of how to develop and implement an algorithm in a new programming language, to synthesis of knowledge pertaining to programming language design and translation. The course addresses the theory of formal representations of programming languages, the major categories of languages that are available, comparative analysis of the implementation of programming constructs, and uses of different types of languages.

Students in the class ( $N = 33$ ) had a mean age of 28.35 years and a cumulative gpa of 3.05/4.0. Since the class was designated as e-Learning, students could not be required to visit campus. The course was comprised of 12 units, each with a number of activities that required students to answer questions, write and/or run program segments and note the results, research answers to open-ended questions, and synthesize ideas from the concepts in the course.

The university licenses Desire2Learn (D2L), a learning management system, and Elluminate, a Web Conferencing software program that has played a major role in the delivery of this course. D2L is a typical LMS that provides a means to create and manage discussion threads, manage and place course content online, establish a grade book, offer quizzes or tests, and provide students with a grade book. A total of six forums were established: one for an introductory post at the beginning of the course, general questions and answers, best times for Elluminate sessions, suggestions pertaining to the conduct of the class, paper-related questions, and programming questions.

#### **Activities**

A total of 12 units covering chapters in an accompanying textbook and other topics of interest were established for the course. Each unit had a number of activities associated with it that ranged from placing pre-defined code into a

development environment and running it, to answering conceptual questions requiring research either online or in the text book. Students kept records of all their work in electronic Activity Logs. Activity logs were submitted weekly and graded. The course materials were established in D2L. Students worked through the materials in order to create their activity logbooks.

Additionally, students learned to program in four languages, corresponding to the four major paradigms of computer programming languages: procedural, object oriented, functional, and rule-based. Assignments, typically based upon development in a new programming environment, were made throughout the semester. Students were also required to create a paper providing an in-depth analysis of a programming language and they took two examinations. While online and open-book, the examinations were timed in such a way that students needed to know the material—it would not be feasible to research questions extensively in real time during the exams.

Web Conferencing learning sessions were conducted each week. Finding the best time for the sessions that accommodated all student schedules was difficult. As a compromise, Elluminate sessions took place at 6:30 pm with sessions alternating between Wednesday and Thursday evenings on alternate weeks. Additionally, sessions were recorded and could be reviewed by the students who do not attend the live sessions. Each session started with a review by the instructor of the highlights of the unit of interest. These discussions helped to lend context to the content and to provide the students with a better understanding of those topics the instructor wished to emphasize. Following this activity, an open question-and-answer session took place. Typically, this component of the Web Conferencing session consumed the majority of the time.

A second major activity in the sessions was a discussion of the programming projects. Students could seek clarifications pertaining to requirements, design approaches, algorithm development, etc. The instructor typically demonstrated how to develop and run a simple program such as “Hello World,” the traditional first program students write when learning a new language. Students were often somewhat overwhelmed when learning both a new language and a new development environment at the same time. Effectively helping students to understand how to use these environments, which is fairly easy to do in face-to-face sessions, is much more difficult to achieve by providing user’s manuals in an asynchronous setting. Additionally, development environments change rapidly over time, so pre-defined materials quickly become obsolete and pose a maintenance problem. The Web Conferencing sessions were by far the most efficient way to help students get started on assignments like these in a distance setting.

### **Outcomes of the Course**

The results of the course were encouraging. Of the 33 students who started the course, 29 completed it. Of the completers, one student received a failing

grade and the rest (28 of the 33) received at least a grade of C–, the minimum necessary for a student majoring in computer science to get credit for the course. This result equated to an 85% overall success rate for the course. The mean gpa for all students except those who had withdrawn was 3.03 on a 4-point scale.

The last two face-to-face offerings of the course conducted by the instructor had overall similar goals and outcomes. They included similar programming assignments and tests of similar difficulty. However, they did not require the weekly Activity Logs. Aggregate outcomes for those two offerings (with  $N=31$ ), included a 78% successful completion rate and an average 3.10 gpa on a 4-point scale. In terms of quantitative results, the online version actually had a nominally higher success percentage with a nominally lower overall average score. Table 1 contains a summary of these results.

Compared with outcomes in other online courses, the success rate in this course was very high. For instance, in the first online offering of this course the previous semester, one in which the Web Conferencing capability was not used, a total of 35 students attempted the course, and 25 completed, a 71% completion rate. This was not a good outcome for a required senior-level computer science course. While the earlier version was conducted by a different instructor and some of the difference was certainly due to that fact, it was also true that students expressed difficulty getting the support needed for the programming assignments.

### SUMMARY AND CONCLUSIONS

The literature suggests that technical courses present special challenges when offered in e-Learning format. It is a particular challenge to teach problem solving skills without efficient instructor-student interactions. Inability to provide effective interactions may lead to outcomes that include increased attrition, and less effective, more idiosyncratic problem-solving strategies. The goal of this article is to identify major benefits and drawbacks entailed by the incorporation of Web Conferencing into technical courses offered in an e-Learning environment. In service of this goal, a case study pertaining to an e-Learning course in Programming Languages that included a Web Conferencing component is presented. Additionally, a new approach to a hybrid course, in which Web Conferencing substitutes for the face-to-face part, is discussed.

Table 1. Outcomes to Face-to-Face and Blended Online/  
Web Conferencing Sections of Programming Languages

Class format	Students	Completers	Percent	Class gpa
Face-to-face	31	24	78%	3.10
Blended online/synch.	33	29	85%	3.03

By any reasonable measure, augmenting the e-learning version of the Programming languages course described in the case study with the Web Conferencing component was a success. Attrition rates were lower than in an earlier e-Learning version of the same course that did not have the Web Conferencing, and equivalent to those of a face-to-face version of the class. The main benefits of incorporating the Web Conferencing were in fostering student learning, understanding, and retention. For example, the introduction of new projects was greatly improved since students could participate either in real-time or through recordings of the sessions in demonstrations of how to use the programming environments, how to tackle relevant problems, etc. The only major shortcomings of the offering of the course described in the case study were the increased demands upon the instructor's time and the fact that the potential of Web Conferencing for assessment activities was not fully exploited.

## REFERENCES

1. J. LeBaron and A. McFadden, The Brave New World of E-Learning: A Department's Response to Mandated Change, *Interactive Learning Environments*, 16:2, pp. 143-156, 2008.
2. Elluminate, Elluminate: Where Bright Ideas Meet. Available online at: <http://www.illuminate.com/>, accessed 6/12/2009.
3. Connect Pro. (2008). High impact web conferencing and eLearning with Adobe Acrobat Connect Pro. Available online at: <http://www.adobe.com/products/acrobatconnectpro/>, accessed 6/12/2009.
4. R. Thomas, Experiences Teaching C++ Online, *Journal of Computing Sciences in Colleges*, 15:5, pp. 214-222, 2000.
5. L. Molstad, Teaching computer programming using distance education technology, *Journal of Computing Sciences in Colleges*, 17:1, pp. 265-277, 2001.
6. J. L. Zachery and P. A. Jensen, Exploiting Value-Added Content in Online Course: Introducing Programming Concepts with HTML and Javascript. *Proceedings of SIGSCSE'05*, Reno, NV: ACM Press. pp. 396-400, 2005.
7. R. Hirshheim, The Internet-Based Education Bandwagon: Look Before You Leap, *Communications of the ACM*, 48:7, pp. 97-101, 2005.
8. J. Gal-Ezer, Online Courses: North Carolina Business and IT Courses—A Case Study, *SIGCSE Bulletin*, 40:2, pp. 18-19, 2008.
9. E. Odekirk-Hash and J. L. Zachary, Automated Feedback on Programs Means Students Need Less Help from Teachers, *The SIGCSE Bulletin*, 33:1, pp. 55-59, 2001.
10. H. C. Lane and K. VanLehn, Coached Program Planning: Dialogue-Based Support for Novice Program Design. *The SIGCSE Bulletin*, 35:1, pp. 148-152, 2003.
11. S. C. Shaffer, Ludwig: An Online Programming Tutoring and Assessment System, *SIGCSE Bulletin*, 37:2, pp. 56-60, 2005.
12. G. Gill, 5 (Really) Hard Things about Using the Internet in Higher Education. eLearn Magazine. Online. Available: <http://www.elearnmag.org/subpage.cfm?section=articles&article=34-1>, accessed 6/12/2009.
13. J. W. Coffey, Digital Natives, Online Learning, and the Production of Capable Computer Science Graduates: The Case for Virtual Synchronous Learning Activities,

- The International Journal of Technology, Knowledge, and Society*, 4:4, pp. 59-64, 2008.
14. M. M. Kazmer and C. Haythornthwaite, Multiple Perspectives on Online Learning, *SIGGROUP Bulletin*, 25:1, pp. 7-11, 2004.
  15. T. L. Friedman, *The World is Flat* (1st Edition), Farrar, Straus, and Giroux Publishers, 2005, ISBN 0-374-29288-4.
  16. G. Ury, A Comparison of Undergraduate Student Performance in Online and Traditional Course, *Journal of Consortium for Computing Sciences in Colleges*, 19:4, pp. 99-107, 2004.
  17. E. M. El-Sheikh, J. W. Coffey, and L. J. White, Exploring Technologies, Materials, and Methods for an Online Foundational Programming Course, *Informatics in Education Journal*, 7:2, pp. 259-276, 2008.
  18. L. White, J. W. Coffey, and E. El-Sheikh, A Longitudinal Study of the Effectiveness of Teaching Introductory Computer Programming Online, *Proceedings of Ed-Media*, pp. 529-536, 2009.
  19. J. Courte, Comparing Student Acceptance and Performance of Online Activities to Classroom Activities, *Proceedings of SIGITE'07*, pp. 185-189, 2007.
  20. J. Kleinman and E. Entin, Comparison of In-Class and Distance-Learning Student Performance and Attitudes in Introductory Computer Science Course, *Journal of Consortium for Computing Sciences in Colleges*, 17:6, pp. 206-219, 2002.
  21. M. Nelson, A. Settle, B. Bhagyavati, D. Shaffer, G. Miles, and J. Watts, Online Teaching Practices (Both Best and Worst), *Journal of Computing Sciences in Colleges*, 21(2), pp. 223-230, 2005.
  22. TurnItIn. Turnitin Digital Assessment Suite. Available online at: <http://turnitin.com/static/home.html>, accessed 6/12/2009.
  23. T. Lancaster and F. Culwin, Towards an Error Free Plagiarism Detection Process, *Proceedings of ITiCSE '01, the 6th Annual Conference on Innovation and Technology in Computer Science Education*. pp. 57-60, 2001.
  24. J. W. Cooper, A. R. Coden, and E. W. Brown, Detecting Similar Documents Using Salient Terms, *Proceedings of CIKM '02, the Eleventh International Conference on Information and Knowledge Management*, pp. 245-251, 2002.
  25. A. Ebrahimi, Problem Solving Variations in an Online Programming Course, *Journal of Educational Technology Systems*, 35:3, pp. 261-279, 2007.

Direct reprint requests to:

John W. Coffey  
 Department of Computer Science  
 The University of West Florida  
 Pensacola, FL 32514  
 e-mail: [jcoffey@uwf.edu](mailto:jcoffey@uwf.edu)



Copyright of Journal of Educational Technology Systems is the property of Baywood Publishing Company, Inc. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.