ABSTRACT
Distance education (DE) is proliferating with no signs of slowing down. This paper aims to fill the gap of lacking literature by providing instructional details of teaching a Web development course in the format of asynchronous DE and offering practical instructional strategies. The unique contribution of this paper lies in exemplifying the online delivery of a highly technical course that has been traditionally taught in face-to-face settings, as well as the application of problem-based learning (PBL) methods to DE.

Categories and Subject Descriptors
K3.2 [Computers and Education]: Computer and Information Science Education – Information Systems Education

General Terms
Design, Performance

Keywords
Distance Education, eLearning, Web Development, Problem-Based Learning, Pedagogy

1. INTRODUCTION
The United States Distance Learning Association (USDLA) defines Distance Education (DE) as “the organizational framework and process of providing instruction at a distance” [24]. Over the past approximately 120 years, DE has evolved from correspondence schools in late 19th century, to the adaptation of communication media (e.g., broadcast TV and radio) in the 1950s, computer-assisted instruction in the 1980s, and Web-based courses in the late 1990s [3,20]. Clearly, technology advance has played a key role in changing the dynamics of DE delivery media [2]. Today’s distance education is predominated by the applications of Internet technologies, and as a result, a new set of terms have been born and used interchangeably, including Web-based DE, Web-based instruction, distance learning, online learning, e-learning, etc. [16]

Due to its extraordinary ability to break through time and space constraints, distance education is proliferating with no signs of slowing down, especially in the higher education arena. The rise in dedicated virtual universities and the growth of online courses reflect this escalation in online education within the United States in particular and around the world in general. The 2010 Sloan Survey of Online Learning reveals that the number of students taking at least one online course in fall 2009 increased by almost one million from a year earlier [23]. The 21% growth rate for online enrollments far exceeds the 2% growth in the overall higher education student population [23].

There is a significant body of research conducted on topics related to distance education, much of which, from the 1980s onward, focuses on the comparative equivalency of DE and face-to-face instruction [3]. As Web-based course delivery has gradually established itself as a legitimate alternative to classroom instruction, and will continue to thrive in modern educational institutions [3], a pressing need exists for a shift in focus from comparison studies to pedagogical studies in the context of DE. In addition, there is an apparent dearth of literature on teaching Web application development in a fully online environment. The vast majority of previous studies cover courses in other disciplines, such as mathematics [27], psychology [9], economics [21], and health science [7]. In this era abounding with various technology-mediated forms of DE, online instruction is no longer limited to text-based discussion boards. IT educators are being challenged to become knowledgeable about effective online instructional methods, as well as how to best use technology to enhance teaching effectiveness.

This paper aims to fill the gap of lacking literature by providing instructional details of teaching a Web development course in the format of asynchronous DE and offering practical techniques for delivering online a highly technical course that has been traditionally taught in face-to-face settings. The rest of the paper is organized as follows. First, the paper describes some special challenges and issues related to teaching Web development at a distance. The next two sections present the instructional strategies and implementation details of the course. The paper concludes with some preliminary findings from the students’ Course Evaluation questionnaire and directions for future explorations.

2. CHALLENGES OF TEACHING WEB DEVELOPMENT ONLINE
Courses on Web development and application programming, which are usually offered by CS, IT, and Engineering
departments, have traditionally been taught in a face-to-face mode and are particularly difficult to redesign for a fully online mode of delivery [5]. Instructors’ general reluctance in delivering such a technical course online was described by Reeves [19]:

I was concerned initially about teaching programming online since I feel that my presence in lecture and lab with the students facilitates the learning process considerably. I did not know how to replace “my presence” in the Web course. (p. 298)

A number of special challenges and issues faced by IT educators exist for developing and teaching Web application development online:

Firstly, due to the technical content of the course, the popular PowerPoint presentation format adopted by many online courses is inadequate for the instruction of developing Web applications. Using online education systems solely for providing or posting static materials arguably adds no real value to student learning [13]. Instructors need to demonstrate explicitly what to do and how to do it in a conversational and interactive teaching environment. However, it is challenging to re-create such classroom experience in cyberspace.

Secondly, it is widely accepted that students learn best by doing, especially when it comes to Web development and programming [26]. In order to develop their own understanding and acquire solid skills, students need to be engaged in hands-on exercises and solving relevant problems through active learning. Such practice is hard to facilitate in class, and more so in an online environment.

Thirdly, most Web development courses require an integrated development environment (IDE) with specialized software and hardware installed, as well as reliable access to a Web server. The technical difficulty of supporting students outside of a classroom or laboratory setting is another cause of concern for online delivery of such courses [5].

Lastly, online classes require a great deal of self-discipline and self-direction. At a distance, it is very easy for students who are poor self-regulators to forget (or claim to forget) to check the course and miss deadlines [5]. This is particularly problematic for courses with heavy workloads such as Web development and programming, as once students get behind in their learning activities, they usually end up with a less than satisfactory grade.

3. INSTRUCTIONAL STRATEGIES

The utilization of an advanced learning management system does not automatically transform a course to an online format or make people better instructors. Effective online instruction requires a high level of responsibility for adopting instructionally sound methods on the part of the instructor, who must balance the requirements of technology, delivery, pedagogy, learning styles, and learning outcomes [10]. In response to those challenges discussed in the previous section, this author has redeveloped a Web development course based on some effective instructional strategies and delivered it in the format of asynchronous DE. A summative description of the instructional strategies and their pedagogical roots is provided next.

(1) Facilitating different types of interaction

Interaction is one of the most critical components of any educational process, and it has long been identified as a defining construct in DE research for its predictive qualities in connection with learning effectiveness and student satisfaction [1,3]. According to Moore (1989), interaction is classified into three types: (1) student-content (SC), (2) student-instructor (SI), and (3) student-student (SS) [17]. To enable students to transform tacit knowledge into explicit knowledge and skills, the DE learning environment should be designed to facilitate all different types of interaction via numerous mechanisms and activities. Although strengthening all three types of interaction seems ideal, different combinations of interaction can be provided in different strengths [1,3].

In the course, SC interaction was supported at the highest level, compared to SI and SS interaction. The entire course was centered on the subject matter under study and targeted at assisting students with the development of knowledge and skills. Specific SC interaction included but was not limited to:

- Reading informational texts (e.g., textbook, code examples, syllabus, PowerPoint lecture slides, assignment specifications, etc.);
- Watching lecture recordings (and following along with hands-on exercises);
- Completing assignments (e.g., online quizzes, labs, projects, etc.);
- Using required software and tools;
- Searching for relevant information and resources

SI interaction helped to personalize the learning and bridge the distance online. A variety of SI interaction in terms of symmetry and synchrony was put to use in the course, including:

- Asymmetrical interaction:
  - Delivering lectures and demonstrating examples;
  - Grading, evaluating, and providing feedback on student submissions
- Symmetrical interaction:
  - Participating in discussion boards;
  - Exchanging emails;
  - Answering questions in chat sessions using texting, audio, and whiteboard

SS interaction took place primarily in the form of discussion boards. As an attempt by the instructor’s to promote an interactive and friendly learning environment, students were first required to post a self-introduction on the Discussion Board, which helped them to become acquainted with each other. Asynchronous discussion activities among students may not only provide opportunities for students to learn reflectively and actively, but also provide them with multiple sources of feedback and help students develop metacognitive and self-regulated learning skills through peer modeling and mentoring [14].

(2) Integrating blended instructional media

The literature is largely univocal about the importance of the variety of instructional media and communication tools required in DE [12]. This is because no single medium can support all
instructional strategies and because a Web-based course must cater to and satisfy students’ various learning styles and needs. The selection of appropriate instructional media is an essential part of developing any course, and a blended approach, which integrates multiple media into instruction, should be adopted for a distance learning environment [12]. The following list represents some primary types of instructional media that were utilized in the course:

- Text (e.g., books, documents, email, threaded discussion boards, Web pages, etc.);
- Graphics (e.g., photos, diagrams, charts, etc.);
- PowerPoint Presentation slides;
- Live chat room supported with texting, audio, and whiteboard;
- Web-based learning management system (using Blackboard®);
- Screencasts with audio narration and video (using Camtasia Studio®) delivered via a streaming media server

It is worth mentioning that screencasting, a method of video recording of computer screen output, is recently gaining popularity among online instructors. Because a screencast captures desktop activity accompanied by commentary, it is particularly effective for explaining computer-based procedures in distance learning situations and creating synthetic environments similar to traditional classroom training [6]. Anything that can be displayed or demonstrated on a computer screen, such as using the integrated development environments, showing PowerPoint slides, and explaining specific coding techniques, can be recorded in the screencast. Theoretically, screencasts should have a positive effect on student learning for its ability to present information in both visual and verbal forms, which is supported by Paivio’s dual coding theory [18] and Mayer’s theory of multimedia learning [15]. The use of Camtasia Studio® in creating screencasts will be further explained in a later section.

(3) Applying problem-based learning methods

Problem-Based Learning (PBL) is a student-centered instructional method of hands-on, active learning in which students learn about a subject in the context of complex, multifaceted, and realistic problems [11]. PBL is in accordance with the constructivist paradigm, which is based on the premise that knowledge is actively constructed by learners based on their prior experiences rather than directly delivered by the teacher [25]. Despite the enormous popularity of PBL across many disciplines in conventional higher education settings, PBL methods are not widely used in the fully online mode. Given the student-centered nature of DE, the Internet is ideal for applying PBL methods to the delivery of instruction since DE courses can exploit the communication features inherent in the Internet to promote active learning that can lead to students’ construction of their own new knowledge and skills. Coupled with tangible goals, stimulating actions, and authentic problems, PBL has proven to be engaging, motivating, and effective [11].

The PBL strategy adopted in the course was carried out in several steps:

Each screencast lecture generally started with a short introduction by the instructor on the particular syntax and semantics that were native to the technologies covered, followed by the instructor’s close examination of some live-code examples and step-by-step demonstration of how to complete specific tasks (e.g., implementing a CSS-based navigation bar, publishing an RSS feed from a Web server, or inserting a Tooltip widget using the Spry framework in Adobe Dreamweaver). Students were expected to follow along with these hands-on exercises.

The instructor often encouraged students to practice what they had just learned by challenging them with a problem to solve, a program segment to complete, or a dilemma to resolve. The challenges raised by the instructor were always based on a concrete problem with some specific, tangible goals, for instance, to add an “onmouseover” event to an image using JavaScript to complete a rollover effect after the instructor had shown how to add an initial “onmouseover” event to the image. Once the instructor presented the problem, the lecture recording was blacked out for ten seconds with only a line of text “Please pause and work on your own” in the middle of the screen, allowing the students to click on the pause button to temporarily stop the lecture recording and work on the problem.

At the end of the ten-second blackout, the lecture recording screen would resume the prior working environment. The instructor would present the solution, usually one of the many ways to solve the problem, and continue with the next demonstration or challenge. Therefore, each lecture consisted of a series of tutorials and problem-solving situations, allowing students to obtain the benefits of active learning tasks. In addition, the lab assignments and projects were closely related to the topics and tasks covered in the lecture recordings and required the development of “real-world” websites and applications. They were used to reinforce students’ problem-solving skills and to further enhance the construction of their new knowledge.

(4) Assessing student performance continually

Assessment of student performance is an essential part of an online course. It’s generally agreed that the evaluation component of online learning should be ongoing and continual [10]. That is, “assessments must be carefully and systematically planned to require students to demonstrate that learning has occurred by completing a specific piece of work at various stages in the course (p. 126).” Continuous assessment allows instructors to keep students engaged in productive instructional activities, to gain insights into student comprehension of the learning materials, and to even identify possible “shirker” students and keep them on track [5,21]. The uses of well-designed rubrics to aid a wide variety of assessments and to provide meaningful and timely feedback are also valued by instructors and students alike [10].

In the course, student performance was continually and thoroughly assessed by the means of five projects with a project presentation, eleven lab assignments, and ten online quizzes. Specifically, the five projects integrated all covered technologies throughout the semester into a final website, with one project building on top of the previous one, and were concluded with a presentation. Almost every lecture resulted in a lab assignment or an online quiz or both, and therefore, students were assessed weekly on their progress on specific learning goals and outcomes. The operationalization of these assessment methods will be elaborated in a later section.
In summary, these instructional strategies make for a suitable and effective approach to teaching Web development in an asynchronous DE mode. The next section discusses the implementation details of the course and the experience gained.

4. COURSE IMPLEMENTATION

4.1 Background

The course, Web Development I (WDI), is one of the three-credit-hour Multimedia & Web Development Concentration courses in our BSIT (Bachelor of Science in Information Technology) program. It has always been offered on campus as a standard lecture/lab course of 15 weeks and taken by junior and senior students enrolled in the program. As part of the ongoing efforts to offer the entire IT degree program online, the decision was taken by this author, who was also the instructor of the course, to redevelop WDI for online delivery. During the Spring 2011 semester, two sections of WDI were offered, one in the traditional face-to-face format, and the other in an asynchronous DE format. Though both sections were taught by the same instructor, they were completely separate, with the DE section delivered in a fully online mode to twenty enrolled students.

WDI is technical in nature and aims to cover the component technologies (e.g., XHTML, CSS, JavaScript, DOM, XML, RSS, XSL) that enable client-side Ajax programming. Students learn how to design and develop advanced websites with interactive interface and functionality. In addition to the standard pedagogical objectives for WDI, a new goal specific to the DE delivery was to demonstrate comparability to the equivalent face-to-face course regarding student learning outcomes.

4.2 Textbook and System Requirements

The textbook used was Internet & World Wide Web—How to Program by Deitel [8]. Fortunately, an electronic version of the book, provided by Safari Tech Books Online, was accessible through the university library website free of charge and was used by most or all of the students in the course. The online book could not be downloaded, but it was only a few clicks away upon students’ logging into the website with their student account.

Students taking the online course were required to own or have access to the following computer hardware and software:

- Access to a configurable and Internet-accessible computer capable of fully running Blackboard® and current browsers with Adobe Flash Player installed and JavaScript enabled;
- Microsoft Expression Web, used as the editor and development environment; students enrolled in the IT program are provided with a free copy through the MSDNAA (Microsoft Developers Network Academic Alliance);
- Adobe Dreamweaver, taught and used in some sessions; a 30-day trial version can be downloaded at Adobe’s website;
- SSH Secure Shell Client (or a secure FTP application), for transferring files to and configuring the student’s Web server account provided by the university; the application can be downloaded from the university’s Technical Support website;
- Microsoft Word (or a word processing application), for preparing some assignment documentations, and PowerPoint, for opening lecture slides and creating project presentations.

4.3 Textbooks and Supplemental Material

Blackboard® (CE8) was utilized as the learning management system (LMS), which can “help instructors to build course materials online and engage with students in an interactive way” [4]. It provided a reliable platform for delivering course content and facilitating communications. Specifically, the following tools were used extensively in WDI:

- Announcements: for listing the learning activities for each week, drawing attention to important deadlines, explaining some specifications from a different perspective, and giving start and end-of-semester remarks;
- Discussion Board: for posting a self-introduction in the first week of the class, asking and answering course-related questions by students and the instructor, maintaining a repository of technical solutions and tips;
- Chat Room: for holding online office hours, interacting via real-time texting, audio, and whiteboard, and enabling private chat sessions;
- Assignments: for listing assignment specifications and rubrics, setting deadlines, collecting student submissions, returning grades and comments, and recording scores in the Grade Book;
- Assessments: for creating online quizzes, setting test conditions (e.g., randomization, duration, attempts, etc.), automatically grading and recording scores in the Grade Book;
- Learning Modules: for presenting course content, linking to other course tools or components, and organizing related course materials in a Table of Contents;
- Selective Release: for controlling the release of assignments, assessments and learning modules;
- Grade Book: for viewing, entering, and managing student grades, calculating a specific grade based on a formula, granting or denying access to the course;
- Calendar: for creating and posting deadlines and release schedules of assignments.

4.4 Computer Lab

4.4.1 Learning Module

The course content was divided into fifteen weekly learning modules, in which the first one was named “Getting Started” and included introductory elements (e.g., the syllabus, an FAQ page, instructions for submitting lab assignments and projects, and a link to the “self-introduction” thread on the Discussion Board), and the last one focused on the last assignment of the course, namely, the Project Presentation. All other learning modules included the learning activities and associated course content for a given week, such as textbook chapters to read, lecture slides to download, lecture recordings to watch, online quizzes to take, lab assignments to complete, and/or projects to submit. Using Blackboard’s Selective Release tool, learning modules were set to be unavailable initially but always released to the students at 6:00 a.m. on Monday morning. One module was released per week, and the lab assignment and online quiz for that module were due by 11:00 p.m. on the following Monday.

4.4.2 Screencast Lecture Recording

Camtasia Studio® [22], one of the most popular screencasting software packages, obtained from the university’s Computing Resources Support, was used to record, edit, and produce the online lecture recording for each week. Creating a screencast
video is simple; all that is needed is Camtasia and a microphone or headset. The software allows users to record the full screen, any part of the screen or a particular application, and to zoom in to great detail, focusing the users’ attention on the important part of the screen and making the text easier to see. Similarly, special visual and sound effects can be added to mouse click and cursor movement to provide additional cues.

It was editing the video that usually took the bulk of the time. Editing functions in Camtasia are somewhat similar to movie editing programs, which can manipulate a number of audio and video tracks, add, cut, or split clips, and insert transitions between screens. In particular, callouts were proven useful for providing explanatory information, highlighting a specific part of a program, or even correcting errors that occurred during recording. However, it is important that callouts should be consistent in terms of shape, color and text size in all lecture recordings. Camtasia also offers a few useful audio enhancements, such as removing background noise and evening out volume levels. At the end of the editing process, a lecture was usually divided into three parts, with each part lasting approximately 15 to 20 minutes.

Once the instructor finished editing the video, it was possible to be converted from the original Camtasia Studio Project file (.camproj) into a variety of cross-platform formats, such as Windows Media video, QuickTime movie, Flash outputs, MP4, AVI, DVD, etc. To streamline the production process and use consistent settings optimized for the intended distribution method, a Production Preset was created to produce a Flash output folder, which included an FLV video in 800 x 450 dimensions and a Flash controller embedded in a Web page for easy Web production. The Flash output folder was then uploaded to the streaming media server maintained by the university, and the Flash controller used was the Blackboard learning module for the respective lecture, effectively preventing access to the files by unregistered students. Therefore, all that was needed for current students to watch the lecture recordings was a browser with the Flash plugin upon their logging in to the Blackboard course folder.

4.4.3 Assessment
Final grades were based on quizzes (20%), lab assignments (35%), and projects (45%). All assessment items were provided with rubrics for clarifying the grading criteria, and students were able to view their grades in the online Grade Book in a timely manner after each submission.

Instead of traditional exams, ten online quizzes were used via the Assessments tool on Blackboard to assess students’ theoretical understanding of the lecture content on the weekly topics. The quizzes were open-book and in the forms of multiple-choice questions. Students were given one attempt and 15 minutes to complete 10 randomized questions in each quiz and could take each quiz at any time during the week in which it was released.

There were a total of eleven lab assignments with the lowest score being eliminated from the final grade. The weekly lab assignments were also designed to put the lecture content into practice, and they were excellent preparation for the quizzes and projects. Students were required to submit the files associated with each lab assignment as a zip folder to the Assignment Dropbox on Blackboard.

WDI assigned five projects that were targeted at the common goal of building a real-world website in iterative and incremental steps. Students uploaded each project to their own Web server account provided by the university and made it directly accessible online. At the end of the course, a project presentation was required for students to showcase their final website in the form of PowerPoint slides with voice narration. The Microsoft PowerPoint narration tool made it easy to record voice narration in each slide with the use of a simple microphone.

5. CONCLUSIONS
At the time of writing the course has been successfully delivered for one semester, and an online Course Evaluation questionnaire was administered anonymously to the students at the end of the semester. The students’ overall ratings of the course and teaching were high (4.88 out of 5) and significantly better than the university’s average. The course ratings were also very close to those of the equivalent face-to-face course (4.86 out of 5), which indicated comparability regarding student learning outcomes. Written comments were especially positive regarding the effectiveness of the lecture recordings and the “cleanliness” of the course organization and requirements. In addition to continuously refining the course curriculum with newly acquired technologies and instructional techniques, future efforts will focus on transitioning to the newer version of the learning management system, Blackboard Learn 9.1, which is to be widely adopted by the university in the upcoming semester. Other areas of prospective exploration may include the adoption of mobile devices and technologies in DE and the accessibility accommodations for online learners with disabilities.

The demand for distance education will only continue to grow. In response to the demand, this paper presents the design and delivery of a Web development course in a fully online setting and fills the gap of lacking literature by reporting the use of a combination of effective instructional strategies in online instruction. The unique contribution of this paper lies in exemplifying the delivery of a highly technical course online and the application of problem-based learning (PBL) methods to distance education. It is hoped that some or all of the instructional strategies presented in the paper will prove beneficial to instructors facing similar challenges in the field of Web Development.

6. REFERENCES


