DOCUMENT RESUME

ED 432 994 IR 019 718

AUTHOR	Powers, Kris
TITLE	A Self-Fulfilling Prophecy: Online Distance Learning for Introductory Computing.
PUB DATE	1999-06-00
NOTE	11p.; In: Spotlight on the Future, NECC '99. National Educational Computing Conference Proceedings (20th, Atlantic City, NJ, June 22-24, 1999); see IR 019 708.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Science Education; *Computer Uses in Education; Conventional Instruction; Course Content; *Distance Education; Educational Objectives; Higher Education; *Instructional Design; Instructional Innovation; *Internet; Introductory Courses; Nontraditional Education; Online Systems; Portfolios (Background Materials); Teaching Methods
IDENTIFIERS	*Course Development

ABSTRACT

This paper describes novel approaches for adapting an introductory computing course for online distance learning, including discussions of the underlying pedagogy and objectives as well as the implementation and results of the online course. The first section describes the original course which was adapted to Internet delivery. The next section focuses on the major objectives and approaches to the online course design. Three major instructional strategies that would be unique to the online version of the course are described, i.e., class cybersociety, online student portfolios, and self-guided online labs. Hardware and software resources are also addressed in this section. The third section describes the course implementation and how well the implementation met the design objectives in terms of student demographics, instructional strategies, and resources. The ultimate goal was to provide an enriched educational experience, subject to economic and related technological limitations. It is concluded that, while the accessibility of the course was successfully maintained, several of the teaching methods meant to enrich the experience were hampered by the shortened time frame in which the course was offered. The course materials are currently in revision for a future offering during a regular 15-week semester. (Contains 13 references.) (AEF)

Research Paper: Curriculum and Instructional Strategies

A Self-Fulfilling Prophecy: Online Distance Learning for Introductory Computing

Kris Powers
Department of Computer Science and Information Systems
Bradley University
Peoria, IL 61625-0167
309.677.2461
kdp@bradley.edu

Key Words: **online courses, distance learning, Web-based education, computing literacy**

Abstract

This paper describes novel approaches for adapting an introductory computing course for online distance learning, including discussions of the underlying pedagogy and objectives as well as the implementation and results of the actual course.

Introduction

Early efforts in the arena of distance learning followed a fairly standard "correspondence course" paradigm: students armed with texts and course notes studied prescribed material, and mailed work back to campus to be evaluated. Letters and perhaps an occasional phone call with the instructor were the only forms of interaction. Because each student took the course independently, there was no notion of a "class" and the time frame for completing course work was arbitrary.

From this starting point, the methods for distance learning have evolved along with technological advances. The mode for content delivery changed as lectures on first audiocassette and then videocassette replaced notes. The advent of telecommunications brought with it a myriad of possibilities, including distance learning in a distributed class format. In this format, students across the globe can be linked as a single class through televised satellite communications (NTU, 1999). Because students collect for class meetings at designated sites over a particular time period, such courses represent a radical departure from the original "correspondence courses" in which isolated students work asynchronously. Unfortunately, such courses also require sophisticated teaching/learning environments, and many institutions lack ready access to the necessary resources.

The current proliferation of the Internet and World Wide Web (Web) has made yet another paradigm for distance learning possible: the online course. In such courses a Web site serves as a repository of course materials, providing access to expository material, interactive labs, and exams. Students interact with both the instructor and classmates via e-mail, bulletin boards, and/or chat rooms (ACCESS/NJIT Distance Learning, 1998). Although the interactivity of such Web-based courses cannot hope to match what is possible with teleconferencing, no special resources are required to offer such a course; any institution with a reasonable Internet connection is suitably equipped.

In this paper we describe novel approaches for adapting an introductory course in computing for online distance learning. These approaches were motivated by two fundamental goals. First, in light of the obvious educational disadvantages of distance learning, we decided that the online elements of the course must effect offsetting educational advantages. This meant creating positive educational experiences not possible in other formats. Second, we wanted the course to be as accessible as possible, keeping the economic and related technological requirements to a minimum. The course should require neither sophisticated equipment nor expensive software; any off campus student with access to a PC, a modem, and a suitable connection to the Internet should be able to participate, for about the same expense as any other course. Our challenge then was to provide an enriched educational experience online, subject to these limitations. In this paper we discuss the underlying pedagogy and objectives of our course design, as well as the implementation and resulting successes and failures of the actual course.

The Original Course

The course we chose to adapt to Internet delivery is an introduction to computing, with no prerequisites. Students study the history of computing, their impact on society, and standard applications, especially those related to the Internet and Web (e.g., e-mail, search engines, and Web navigators). Students are also exposed to basic computational problem solving using, for example, the BASIC programming language. Current on-campus versions of the course are taught in a discussion/lab format. Discussions address historical and societal issues, and provide an opportunity for the instructor to introduce technical subject matter. The technical subject matter is extended through instructor guided, hands-on experience in the labs. Student work in the course typically includes written themes on societal aspects, the creation of Web pages, assorted artifacts of Web browsing (e.g., electronic postcards to the instructor, links to credible Web reference material, etc.), and programs.

The course is also distinguished as a junior level elective serving a bimodal student population. One population taking the course is Computer Information System (CIS) majors for whom the course satisfies an upper level elective requirement. The second population consists of non-majors, for whom the course fulfills a university-wide "Science and Technology" requirement.

Design of the Online Course

In this section we describe our major objectives and approaches to the course design.

Instructional Strategies

In considering how the Internet could be used for distance learning, we identified three major instructional strategies which would be unique to the online version of the course, and which held the potential for a quality educational experience not otherwise possible.

Class Cybersociety

One of the major societal impacts of the Internet is the development of online "cybersocieties," societies defined in and by a virtual environment (Bruckman, 1996). Students in the course would develop an understanding of the effect of cybersocieties on human interactions by actually participating in one. For instance, through online class discussions students would experience firsthand how the anonymity factor inherent to this kind of communication effects their interaction (Turkle, 1996). In fact, each student's class identity would be established entirely online. (Unlike an on-campus exercise to simulate the effect.) Class interactions with the professor and classmates would only occur via e-mail, bulletin boards, and (at scheduled times) meeting in chat rooms. (A bulletin board is an asynchronous form of communication, in which each participant reads messages and posts their own replies at different times, according to their own schedule. In contrast, a chat room supports the synchronous exchange of messages among participants.) Because fostering a vital electronic community would seem to be better facilitated by intense involvement, we decided to run the course in a shortened (five-week) time frame, similar to the summer school sessions at many institutions of higher learning.

Online Student Portfolios

Students would not just create a page or two, but develop an entire electronic portfolio as a Web site. Each individual student portfolio would include links to papers, programs, and other assigned Web pages. Students would access each other's sites for peer review and as a basis for discussions. A student's portfolio would not only establish their identity in the class cybersociety, but would also be a significant element of their course assessment. The additional benefits afforded by portfolio assessment are well documented (Adams & Hamm, 1991; Hart, 1994).

Self-Guided Online Labs

Using computing as the instructional medium for a course about computing seems exceptionally appropriate. Indeed, the students' experience with the technology that supports the online course (e.g., standard Internet applications) is part of the requisite subject matter. This primary objective of the course becomes a self-fulfilling prophecy. Initial lab work must facilitate students downloading and installing any needed software. Further labs must address the variety of Internet tools that are used. For example, in addition to the class interactions previously described, students will download software across the Internet, use search engines, and navigate to class notes and other materials on the course Web site.

Resources

In keeping with our goal of accessibility, we decided that the course would be offered requiring as little computational power as feasible. Obviously, taking a course online requires a computer, a modem, and a connection to the Internet. For ease in initially developing the labs, we decided that we would start by supporting only the Windows platform. The requisite software will dictate the requisite computational power of the machinery. For the economic considerations previously mentioned, all course software should be either commonly available (e.g., word processor, spreadsheet, etc.) or be in the public domain. We selected Netscape Communicator for Internet software services, including browsing, e-mail and FTP, as well as for HTML editing. A more difficult issue was that of the computational problem solving aspect of the course. This portion of the course had frequently been taught using languages that required access to departmental machines with the necessary translators. However, one version of the course was notable in that it had used a programming language that avoided this requirement: the PostScript printer programming language.

All of the basic constructs of computational problem solving are readily accessible in the PostScript language. About one third of the PostScript language is devoted to graphics. The remainder makes up an entirely general purpose programming language including variables in the most common data types (reals, booleans, arrays, and strings), procedures, and control structures like loops and conditionals (Reid, 1988). While application programs usually generate PostScript programs, these programs are ordinary text files of printable ASCII characters, and can also be generated by human programmers. To view the results, neither a proprietary translator program nor a PostScript printer is required. Students can examine the output graphical images using public domain software like Ghostscript/GSview.

Finally, while learning the PostScript language is arguably of little utility to students wanting to program, learning to program in any language is not an objective of this course; too little time is available to achieve that goal. Rather, students are merely intended to develop an appreciation for the complexity of software development, and the difficulty of producing error free programs.

Implementation of the Online Course: Successes and Failures

The course was offered for the first time in a five-week summer session of 1998. In this section we will describe the more salient features of the course implementation, and how well the implementation met the design objectives.

Student Demographics

Ten students initially enrolled in the course. Three of them were computing majors (computer science, computer information systems, or business computing) and one was a university employee in computing services. Nearly all were already well versed in word-processing, spreadsheet, and graphics programs. As would be expected, the computing majors also had the benefit of previous programming instruction, and various levels of experience writing html, administering Web sites, and using Internet applications. While several students were actually on campus or working in town during the course, none of these students had face-to-face or telephone contact with the professor during the course. Of the 10 students initially enrolled, six of them eventually completed the course.

Instructional Strategies

Class Cybersociety

The course bulletin board was our primary vehicle for class communication. Each discussion concerned a specific topic and lasted for a certain time frame. For example, the discussion on technical issues was ongoing throughout the semester, while each discussion on a particular reading assignment lasted only a few days. The topic of each discussion was introduced by a base message posted by the instructor. (An excerpt from such a base message for a discussion on a reading assignment appears in Figure 1.)

> In an online course, where we are dependent upon our computing resources for our course work, it seems appropriate to begin our consideration of the societal impact of computing with issues of reliability. A famous computer scientist Edgar Dijkstra once stated: "Testing can reveal the presence of errors, but can never guarantee their absence." … What, if any, standards of reliability can be enforced? Should reliability have to be approved by a government agency before release? If so, should it be necessary for all systems or only "critical" ones? And just what should be considered "critical?"

Figure 1. An excerpt from a discussion base message

Student participation in these discussions accounted for 20% of their course grade, and their postings were evaluated according to the criteria summarized in Figure 2.

Unfortunately, the creation of a dynamic class cybersociety was hampered by two problems that ultimately plagued much of the course, in some fashion or another. First, while enrollment was limited to avoid instructor overload in these experimental courses, it did not account for the higher attrition rate generally experienced by these types of courses. The eventual class size ended up too small. The second problem stemmed from the combination of the time frame for the course and student expectations for the distribution of their time commitment. Student motivation for taking this or any online course is generally the freedom it provides from the time and place restrictions of class meetings.

Topical A good message should be relevant to the current subject matter.
Concise Messages several screens long do not get many replies - attempt to use a single screen.
Provocative A good message is one that prompts others to reply.
Substantial A good message either poses an interesting question, or provides a knowledgeable answer which explores, explains or expands a concept.
Timely Replies to other messages must occur in a timely fashion for an effective discussion to occur.
Logical A good message that is not a question should contain a clearly stated thesis supported by factual evidence and valid inferences.
Grammatical The use of grammatical English is especially important for our (potentially) international audience.

Figure 2. Evaluation criteria for bulletin board postings

However, there are different degrees of freedom possible. A course may be entirely asynchronous, allowing each student to proceed at their own pace. Or it may be synchronous, and force students to operate within strict time frames. Clearly, a dynamic, interactive course with class discussions requires the latter. In a course taught in a shortened, summer school session, these deadlines must necessarily occur at a fast and furious pace. A truly interactive environment would require every student to participate on *at least* a daily basis. This conflicted with most of the students' expectations for the degree of scheduling freedom allowed by the course.

In fact, most students had travel or other commitments that kept them offline for days at a time. Even under normal circumstances, most seemed to log onto the system only immediately before scheduled deadlines. The result was that each student usually made only one posting to a particular bulletin board discussion, and this posting often responded only to the base message, as though it were a singular essay question. Further, student time commitments made the scheduling of regular online class chats impossible. In short, the course goal of interactivity failed.

Another difficulty in achieving the envisioned class cybersociety was also experienced by Amill, Harris, Jones, and O'Bryan (1997) in their effort to network educators and experts. In their work, the authors found that electronic mail "requires somewhat different interaction strategies if it is to be used to create maximal educational benefit ...." The same is equally true for any written, electronic communication. The critical role of a discussion moderator, particularly to challenge ideas and force interactions, cannot be overstated.

This aspect of the course, however, was not entirely without success. Additional inter-student class communication was supported by a student contact page which provided classmates with each other's e-mail address and home page URL. This mechanism led to spontaneous occurrences of collaborative learning, with computing majors coming to the aid of the other students in the course. While students could have solicited this aid either privately from the instructor through electronic mail or publicly from the instructor and/or other students through the bulletin board, they were largely more comfortable seeking out each other's help individually.

Online Student Portfolios

Each student created a Web site as an electronic portfolio of class work which was assessed 1/2 on its presentation as a Web site and 1/2 as a portfolio of intellectual development (as related to course content). Like the course discussions, this work accounted for 20% of each student's course grade. Also like the course discussions, the development and use of the portfolios was hampered by the shortened format of the course. A "portfolio documents learning over time. It is this long-term perspective that makes portfolios such a useful assessment tool" (Hart, 1994). While the specific duration of time necessary to achieve this benefit from portfolios is not clear cut, the five weeks allotted to this courses seemed insufficient. Further, the Web sites were not substantially in place until it was too late for their envisioned use as individual identities within the cybersociety. Finally, there were significant student concerns over this type of assessment. While students were provided with descriptions and examples, these seemed insufficient to allay their uneasiness. In the future, we hope to provide explicit, step-by-step guidelines for the development of this documentation.

Self-Guided Online Labs

Various homework assignments throughout the term, required students to download, install, and use course software. Online tutorials for accomplishing these tasks were collectively provided by the course Web site, the university's computing services Web site, and other dedicated sites (Bradley University, 1998; NCSA, 1998; Using Composer, 1998). The public domain software finally used included: WinZip, Netscape

Communicator, QVT/Term, and Ghostscript/GSview. As was indicated by the course demographics, the students drawn to this course were already computer savvy. In light of this fact, judging the effectiveness of these labs and technical support in any meaningful way is not really possible. Only one student, a computing major, reported encountering any difficulties in installing or using course software.

Resources

The technical support of the course was provided through usual campus computing services as well as our Center for Asynchronous Learning which supports a beta implementation of the CyberProf software developed at the University of Illinois, Urbana-Champaign. This software provides an environment for the development and integration of World Wide Web teaching resources, including Web page templates, bulletin board and quizzing facilities, and grade keeping features. In addition to discussions, homework, and portfolios, we had intended to assess student performance using quiz items developed in CyberProf. Unfortunately, this part of our installation of the software package has been bug-ridden and was unavailable for use. When the full release of CyberProf becomes available, we expect that the creation of these items will be a significant part of this course's continued development.

As was previously discussed, the PostScript printer language was used for teaching computational problem solving in the course. Homework assignments ranged from modifying existing programs to writing small programs from scratch. Online tutorials for using the PostScript language were, like the other described lab work, supported by the course Web site and other dedicated sites (Weingartner, 1997). While the primary motivation for using this particular language was to minimize the required resources, in the end analysis the use of this language was highly successful for a couple of additional reasons. First, the graphical nature of the output (e.g., a box) seems much more gratifying to many students than textual results (e.g., Hello world!). Visual learners are particularly aided by this approach. Also, PostScript is an interpreted, stack-based language similar to FORTH; Figure 3 gives an example program and Figure 4 shows the output produced.

```
%!
% This program repeatedly prints the
% string "NECC" in increasingly
% darker shades of gray.
%-----------Procedure-----------
% This procedure prints the string
% "NECC" at the current origin.
/printNECC
  { 0 0 moveto (NECC) show} def
```

Figure 3. An example PostScript program

(figure continues)

```
%-----------Main Program-----------
/Times-Italic findfont 30
scalefont setfont

% move the origin of the coordinate
% system to the middle of the page
320 400 translate

% start, increment, and end values
% for the loop counter
.95  -.05 0 {
  % set gray based on the loop
  % counter on top of the stack
  setgray
  % print the string "NECC"
  printNECC
  % move the origin of the coordinate
  % system up and to the left
  -1 .5 translate } for

% print "NECC" one more time,
% this time in white
  1 setgray printNECC
showpage
```

Figure 3. (continued)

Figure 4. The output of the example PostScript program in Figure 3

This programming model is quite different from the C++ world in which our computing majors do the bulk of their course work. This fact has a couple of happy consequences. Often the computational problem solving portion of this course suffers from the bimodal nature of the student population. Either the computing majors in the course end up bored, or the other students end up confused. Because PostScript is so dissimilar to our typical computing major's previous experiences, this problem is avoided. Further, our majors have the added benefit of extending their experience base in programming languages.

Conclusion

We have discussed the underlying pedagogy and objectives as well as the implementation and results of an online course in introductory computing. Our ultimate goal was to provide an enriched educational experience, subject to economic and related technological limitations. While the accessibility of the course was successfully maintained, several of the teaching methods meant to enrich the experience were significantly hampered by the shortened time frame in which the course was offered. Currently, the course materials are in revision for a future offering during a regular, 15-week semester. The course will also be more actively marketed to draw in the computing novices who have the most to gain from the self-fulfilling prophecy of an online course in introductory computing.

References

ACCESS/NJIT Distance Learning. (1998). [Online] Available: http://www.njit.edu/njIT/Directory/Admin/Distance_Learning/.

Adams, D. & Hamm, M. (1991, May). Portfolio assessment: It's not just for artists anymore. *The Science Teacher*, 18-21.

Amill, L., Harris, J., Jones, G., & O'Bryan, E. (1997). Simple, but not easy: Practical lessons in telementoring. *Proceedings of the National Educational Computing Conference*, Seattle, Washington.

Bradley University. (1998). How to create a home page. [Online] Available: http://www.bradley.edu/help/homepage.html.

Bruckman, A. (1996, January). Finding one's own space in cyberspace. *Technology Review*, 48-54.

California Virtual University. (1998). [Online] Available: http://www.california.edu/.

Hart, D. (1994). *Authentic assessment*. New York: Addison-Wesley Publishing, Inc.

National Technological University (NTU) Home Page. (1999). [Online] Available: http://www.ntu.edu/.

NCSA—A Beginner's Guide to HTML. (1998). [Online] Available: http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html.

Reid, G. (1988). *PostScript language program design*. New York: Addison-Wesley Publishing, Inc.

Turkle, S. (1996, February/March). Session with the cybershrink. *Technology Review*, 41-47.

Using Composer. (1998). [Online] Available: http://help.netscape.com/docs/client/pe/asw/reflib/introcom/chap05.htm.

Weingartner, P. (1997). A first guide to PostScript. [Online] Available: http://www.cs.indiana.edu/docproject/programming/postscript/postscript.html.

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").