Evolving an Integrated Electronic Environment

For Asynchronous Delivery of CS1

Wendy A. L. Fowler
Richard H. Fowler

Department of Computer Science
University of Texas - Pan American
Edinburg, TX 78539-2999
(956) 381-2320
wfowler@panam.edu

ABSTRACT

The advent of low-cost, multimedia computers and the development of affordable global and local networks, provides new mechanism for support of teaching and learning. Students gain increased access to teaching and learning resources and the quality of learning experiences can be enhanced. The educational community to create an environment for the delivery of all basic resources for learning provision (lecture, practical exercises, assignments, and evaluation tools) and where students can gain skills needed for the pervasive changes in information access and utilization that are occurring. These skills center on developing techniques for learning and working in an environment that has a broad range of heterogeneous information resources. Based on the educational theories of collaboration and constructionism, the learning and teaching environment we evolved provides students opportunities to acquire skills needed for success in a distributed workplace in which electronic tool use and information discovery, organization, and utilization are the principal activities in problem solving.

INTRODUCTION

This paper reports on the evolution of an approach to teaching an introductory programming course in a highly integrated, electronic environment. CS 1380: An Introduction to Computer Science is an introductory computer science course requiring the development of applied skills to begin programming as well as providing a theoretical foundation for continued study. Students learn general problem solving skills, algorithm design, a programming language, and are introduced to the principles of software engineering. Maintaining a balance among these somewhat disparate skills is one principal challenge for both instructor and students. A second challenge lies in preparing students to apply these skills in a workplace using tools to apply and manage resources. The integrated electronic environment is designed to address these challenges.
The motivation for our implementation is as follows: Since a diverse set of skills is required and the potential work environment has a broad range of distributed heterogeneous information resources, students should acquire the skills necessary for success in a distributed workplace where electronic tool use and information management are the principal activities in problem solving. If the work world demands that they be situated in self directed activities of making, then learning should be situated in the same activity. If the work world demands that they work collaboratively, it is essential that students learn to collaborate as well as learn through collaborative activities. Focus is maintained in the environment on information exploration, discovery, manipulation, and integration. Additionally, small-scale communities, within which students work, are created by providing an intensive electronic environment and structuring student activities through electronic communications. Thus, tools that coordinate information access, pedagogy, problem solving, and design form the core of the student's learning and working environment.

The Programming Design and Programming (PDP) environment we create combines a pedagogic CASE tool, a programming environment, web-based hypermedia, indexing tool, group oriented laboratory exercises, and synchronous and asynchronous communication mechanisms as instructional tools [5]. Students access WWW materials and use the visually based CASE tool (Design Tool) and programming language in program design and implementation. Asynchronous workgroups are used in team software engineering projects facilitated through newsgroups serving as repositories for group code and reports. The PDP environment is an evolving environment. Technical innovations are implemented as they arise and the whole system is in a continual process of improvement.

THE EVOLUTION OF THE PDP ENVIRONMENT

The PDP environment evolved over a six-year period of time. During the 1993 / 1994 academic year a notebook written in Owl’s Guide was released as a very richly linked hypermedia document to supplement the traditional classroom environment. Design Tool, a pedagogic CASE tool, was designed and implemented for use in conjunction with the notebook during the following year. To improve student accessibility and to take advantage of the growing enthusiasm for the WWW, the hypertext notebook was migrated to the WWW during that second year. The first version of the notebook was a rather flat set of web pages, and rather disappointing in contrast to the stand-alone hypermedia version. A second WWW version was released later that took advantage of frames and provided a richer environment, albeit, still not as rich as the original notebook. Lab exercises were developed and during 1995 course delivery moved from a traditional approach to one which integrated a set of individual and group oriented laboratory exercises in place of one third of the lectures. Discussion groups were added to facilitate the lab exercises and the exchange of ideas between general class members and between group members.

DUSIE (Dynamic User defined Searchable Index Engine), an indexing tool for Web documents, was designed, implemented and tested with the rest of the tool at the end of 1995. The complete tool set was introduced as an integrated environment to augment
student learning in 1996. During 1997, the PDP environment was used to both augment and provide flexibility in traditional course delivery. In experimental sections of the course, the meeting time for the lecture component of the class was not mandatory. Students worked through the course content based on the schedule provided in the syllabus on their own and attended lecture when they felt they needed additional assistance with a concept. However, student lab attendance remained mandatory.

With the adoption of an authoring system for online course delivery on the University of Texas-Pan American campus, the hypertext notebook was migrated again during the summer of 1998. The web documents were loaded into an online authoring system (WebCT) that provided online course administration tools for both students and the instructor. Incorporating the content into the online delivery system allowed the notebook content to more closely resemble the original version of the hypertext introduced in the fall of 1993. The complete set of integrated tools was used for online delivery of CSCI 1380: Introduction to Computer Science during spring 1999. The notebook was revised during fall of 1999 and is in use to support a competency based version of the course.

PDP ENVIRONMENT: NETWORKED WORKSTATIONS AND MULTIMEDIA

The PDP learning environment utilizes networked workstations and web-based multimedia to prepare students for self-directed learning in a distributed work environment. Student activities are structured through laboratory exercises and electronic communication leading to the formation of small-scale communities in which students work on tasks using software engineering techniques for program development. Given course content and industry practice, extensive use of electronic tools for distributed software development is natural.

The environment provides students with an integrated suite of pedagogically oriented electronic tools and heterogeneous information resources. The tools and the interrelationships among the tools support independent development of each of the skills necessary for programming and managing information resources, while keeping more abstract concepts of computer science in sight.

GOALS AND ACTIVITIES IN THE LEARNING ENVIRONMENT

The design goals and activities of the PDP environment are based on a framework that combines ideas of constructionism and collaboration, and the skill requirements of the 21st century workplace. The goals concentrate on fostering a general orientation to learning that is self-directed, active and group-oriented. The activities are drawn from discussions of the future of the workplace and society [2,3,8], other's experiences in implementing such learning environments [1,4,9,10,11], our own experience using hypermedia in our classes [6], and the unique needs of our student population.

Activities focus on collaborative work relations, electronic tool use, information access and utilization, and efficient use of human resources. Collaborative communities are created through synchronous and asynchronous communication within and between groups where individuals’ roles change from learner to mentor to collaborator. Pervasive
use of integrated electronic tools leads to the students acquiring skills for the facile manipulation of diverse information resources where iterative searching and collating of information, information reuse across time and domains, and joint information use are necessary.

To address this set of design goals and activities we create a distributed, highly integrated electronic university learning environment where interaction among students. Learners acquire knowledge by engaging in self-directed learning activities and in group oriented, collaborative projects. The following sections outline the components of the PDP environment.

**HYPERTEXT NOTEBOOK**

The web-based hypertext serves to integrate concepts introduced in the classroom with the design and programming environment. The notebook contains the course lecture material with static graphics, animations of control and data structures, and course supplements including course syllabus, schedule, laboratory exercises, programming assignments, and a large set of designs and sample programs. The hypertext is structured into modules that reflect the sequence and scope of lectures. Within modules, main lecture topics serve as organization points from which students can explore concepts in greater depth. Each module provides numerous samples illustrating newly introduced concepts that build on previous samples and concepts. Students can experiment by copying programs into the programming environment, compiling, and executing the code. All ‘lecture’ examples are included in the hypertext.

**LABORATORY EXERCISES AND PEER TUTORING**

The laboratory exercises provide the unique benefit of bringing people together. Coupled with chat and threaded discussion groups, individual and group lab work allows students to engage in informal discussions with faculty and peers. Students gain new perspectives. Group and project management lab exercises help students learn about collaboration, as well as learn via collaboration. Through these group activities students gain experience in coordinating efforts with others whose skills and knowledge may differ greatly from their own. Combined with threaded discussions and chat rooms, the collaborative exercises serve to support peer mentoring and tutoring.

**ELECTRONIC COMMUNICATION**

Collaboration is emphasized through group learning in a virtual community of computer science learners communicating and sharing information even when geographically separated. Local newsgroups foster the emergence of these virtual communities. Students have a place to discuss problems as a group at their own pace. Students are required to contribute to ongoing, monitored discussions. The newsgroups provide an element of immediacy, or at least relevance. The archival features of the newsgroup allow the electronic conversations to persist through time. This supports the
formation of a community by eliminating the temporal constraint of face-to-face conversation.

These newsgroups work well because they support asynchronous learning and work, required in an online course. The newsgroups provide an element of social interaction in electronic space. Online learning is not a solitary experience, but rather occurs in a social context which includes information exchange, collective decision making, conflict resolution, and peer teaching [12,13].

Private communication between participants (students and instructors) is facilitated with e-mail. The use of e-mail provides a different structuring of the instructor-student relation: some of the practical difficulties of providing rapid feedback on questions are overcome. Moreover, students are relatively thoughtful in framing their questions. Student messages tend to be brief and focused, and the faculty is able to respond at the appropriate skill and knowledge level.

DESIGN TOOL: A PEDAGOGIC CASE TOOL

Design Tool provides a visually based system for problem solving. Using problem decomposition and solution design via structure charts, data flow checking, Pascal code generation, and report production [5] students learn how to design solutions. Since designs to interesting problems can become quite large, an overview diagram of the entire structure can be used in orienting the displayed modules to the complete program design. The overview also provides navigation facilities for moving within the design's module structure. Display facilities for panning, using the overview, and zooming in and out on parts of the structure are intended to help students keep track of the overall design during development.

DYNAMIC USER INDICES

As described so far, the PDP environment provides a rich assortment of resources to support information retrieval and understanding. The Dynamic User-defined Searchable Index Engine (DUSIE), allows students to capture and organize meta-knowledge about the resources. Students construct a personal repository of concepts and concept relations to complement the author centered hypertext document structure [7]. As an extension of the virtual classroom, the tool reinforces learning through the active construction of an associative network of terms and annotations. Students create indices into the hypertext documents that explicitly define their understanding and use of the concepts. They impose their ordering on the concepts and modify their ordering as their perceptions of the concepts change. Therefore, the index reflects the student's conceptual domain and can be used efficiently and effectively to retrieve information from the documents.
EVALUATION OF THE PDP ENVIRONMENT

A preliminary evaluation was conducted in spring of 1996. All components, as described above, were used in three sections of the target course. Throughout the semester, components were used in the classroom as the basis for presentation and in a formal lab setting for hands-on instruction. Standardized assignments, including design and implementation elements, were used to test the integration of the tools into a learning environment.

During fall 1996 and spring 1997, twelve (12) sections of the target course were selected for assessment. Of the six designated sections each semester, three sections used the traditional learning environment while the other three used the integrated learning environment. A series of lab and design exercises, programming assignments, and exams for use during evaluation. The number, timeliness, quality of programming assignments completed, and exam performance was tracked. The number, timeliness, and quality of lab solutions were tracked in the PDP environment.

Overall student response to the learning environment was positive. Noticeable improvements in students' ability to design solutions to problems and implement programs occurred. As would be expected, overall computer lab utilization increased with the use of the PDP environment. However, the increase in lab use was not due to an increase in time spent programming. Actual programming time decreased. The increase in time was associated with increased student engagement with the resources and experimentation. Roughly 20% of the programs included design or programming concept enhancements discovered through courseware exploration.

Results of the preliminary evaluation indicated a lower drop rate and a higher success rate as evidenced by a grade of C or better in the PDP environment versus the traditional teaching style. Table I. summarizes the final grade distributions for the programming course offered Spring Semesters in 1994, 1995, and software-aided course offered spring 1996.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9%</td>
<td>11%</td>
<td>15%</td>
</tr>
<tr>
<td>B</td>
<td>11%</td>
<td>11%</td>
<td>22%</td>
</tr>
<tr>
<td>C</td>
<td>14%</td>
<td>11%</td>
<td>22%</td>
</tr>
<tr>
<td>D</td>
<td>11%</td>
<td>3.5%</td>
<td>3.5%</td>
</tr>
<tr>
<td>F</td>
<td>6%</td>
<td>3.5%</td>
<td>3.5%</td>
</tr>
<tr>
<td>DNC</td>
<td>49%</td>
<td>60%</td>
<td>33%</td>
</tr>
</tbody>
</table>

Table I. Final grade distributions as a percentage of the total grade distribution for spring 1994, 1995, and 1996. (DNC - did not complete course)
During the formal evaluation, student response to the system was positive. The effectiveness of the integrated electronic environment was evaluated through comparison of student performance and outcomes in course sections using the standard environment and those using the PDP environment. As discussed above, student performance was tracked using a defined series of assignments, exercises and exams.

The final grade distribution for the designated sections of the target course is presented in Table II. Seventy one percent (71%) of the students using the PDP environment completed the course while only 54% of the students enrolled in the sections using a more traditional delivery completed the course. Additionally, 70.5% of the students completing the course did so with a grade of C or above using the PDP environment while 54% of the students in the traditional sections earned a C or better.

Table II. Final grade distributions as a percentage of the total grade distribution for fall 1996 and spring 1997 for experimental sections by environment type.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Grade</th>
<th>&lt;C</th>
<th>Did Not Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>A/B 22.9%</td>
<td>12.7%</td>
<td>18.5%</td>
</tr>
<tr>
<td>Integrated</td>
<td>41.3%</td>
<td>20.7%</td>
<td>8.5%</td>
</tr>
</tbody>
</table>

A finer level of evaluation was completed which encompassed the quality of programming assignments in both delivery modes and timeliness and quality of laboratory exercises assigned in the sections using the PDP environment. While students in both environments achieved a comparable level of quality in their solutions to programming problems, a greater number of students reached levels affording them a final grade of C or better in the target course when using the PDP environment. Based on the range of scores for lab quality and qualitative assessments made by students, the labs afforded students an opportunity to become both comfortable and agile in the PDP environment. It appeared that the lab experience, while increasing student workload, did not have a detrimental affect on course outcome. Rather the laboratory enhanced the benefits provided by the PDP environment as reflected in the 26% increase in number of students achieving a C or better in the class.

During the 1997/1998 academic year, we continued the on-going process of implementing technical innovations, using the environment in our daily teaching, exploring new concepts and teaching methods, and evaluating the whole setting. During spring 1999, we decided to try using the environment as an on-line course. Experimental in nature, many students were not familiar with the nature of an online course and signed up out of curiosity. Unfortunately, about ¼ of the students were not prepared to take an online course and dropped the course before the first exam. Of the 25% that remained in the class, 100% completed all of the coursework and earned a grade of C or better.
Table III. Final grade distributions as a percentage of the total grade distribution for completion of the online course using the PDP environment.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Grade</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PDP</td>
<td>A/B</td>
<td>C</td>
<td>&lt;C</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>75%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Although the results of the experiment were somewhat disappointing with respect to overall retention, student performance for those who completed the course was above that in both the traditional classroom and a highly mediated environment. In looking more closely at student progression through the courseware, quality and timeliness of lab and programming assignments, exam performance, and student and instructor feedback, changes will be made with respect to the pedagogic methods used during course delivery. Students will be provided with support materials that deal with taking online courses separate from materials dealing with subject content. Students will be encouraged more aggressively to use chat facilities for mentoring and a portion of their grade will be determined by their participation in threaded discussions. Self-checking quizzes will be redesigned to better reflect the type of learning that is required to be successful in the course.

Finally, we decided that the instructor for the course should be one the principle developers of the PDP environment or someone using the PDP environment to augment a traditional classroom. The lesson learned through the online experience is in part the impact of instructor familiarity with the materials on the delivery of the course. The instructor for the experimental section had looked at the materials during the previous semester, but did never used the PDP environment in a highly mediated fashion. In retrospect, the instructor decided that he would have liked to customize certain aspects of the course (quiz content and some programming assignments). In the future, others wanting to teach an online course using the PDP environment will go through a transitional process including a highly mediated version of the course rather than going directly online.

CONCLUSIONS

We describe a university learning environment in which students can develop the knowledge and skill base required in an information-intensive workplace. In this project we use technology to implement educational strategies that include collaborative, project-based learning in an information-rich, tool-rich environment where both synchronous and asynchronous communication can occur to improve student-faculty and student-student interaction. We believe that these strategies influence the course of students’ development and better prepare them for entry into today’s workplace.

The overall experience of working in a highly distributed, yet integrated, environment is valuable for the students: students gain insight and skills that they might
not otherwise acquire. Learning is enhanced because it is immediately supported by practical, hands-on application of theory and guidance by faculty and peers. Students construct knowledge through collaboration as well as learn about collaboration.

Finally, it is important to remember that information technology is the medium to enable changes in the curriculum and not the curricular content itself in this case. The technology supports active and collaborative learning by facilitating work on complex projects, encouraging the rethinking of assumptions, and allowing synchronous and asynchronous discussions. Technology supports the evolution of students into life-long learners.

REFERENCES


**ENDNOTES**

1. Students introduced to the environment spring 1996 integrated several of the tools into the working environment for subsequent courses.

2. An additional 18.4% students achieved either an A or a B and an additional 8% achieved a C using the PDP environment.