New Pedagogical Approaches to Computer Science Education: A Case Study in Peer Learning

Walter R. Rice, Undergraduate, Department of Computer and Information Science
Yasmin Beg, Undergraduate, Department of Electrical Engineering
Carole Waters, Undergraduate, Department of History
M. Krimo Bokreta, Dean of Kings Court/English College House
Jorge J. Santiago-Aviles, Department of Electrical Engineering
University of Pennsylvania
Philadelphia, PA 19104

Abstract *– We believe that a peer community focused on science and technology fosters the development of high-technology knowledge within the constantly changing field of computer science. Within this environment, each student can gain additional insight and practical knowledge, making peer learning an invaluable addition to a traditional curriculum. This paper examines both formal and informal computer science education within the Science and Technology Wing of the University of Pennsylvania.*

Introduction

A simple look at a student weekly schedule reveals the amount of time that a student spends outside the classroom. One can easily assume that the largest portion of that time is spent in interactions with their peers. And if we accept that the student learning process happens continuously in the student life phase, then the nature of the peer interactions become a crucial and a fundamental asset in the students' developments **[1]** and profoundly affects the students' academic performance **[2]**. The true peer learning influences happen along three main axes of interactions: Peer Tutoring, Cooperative Learning, and Peer Collaboration **[3]**. Therefore, peer learning can be used as a strategy to enhance student learning, concept development, and problem solving skills. The attempt of this paper is to analyze some of the parameters and factors that characterize one of the learning initiatives within STWing **[4]**. This paper looks at programs that have been developed entirely by students for themselves and their peers, supported by an environment conducive to extra-curricular academic endeavors.

The Science and Technology Wing

The Science and Technology Wing, otherwise known as STWing, is a living-learning program at the University of Pennsylvania. As such, its goal is to provide an informal residential environment that encourages learning. This takes the form of a two-fold approach, on one hand attempting to increase student-faculty interaction and undergraduate research, and on the other, promoting the benefits of a strong peer community and providing resources essential to its success.

The particular focus of STWing over the 10 years of its existence has been the shared interest of its members in computer technology and networking. In 1989, 12 students were offered room with the first residential Internet access on campus, and formed a core group that has expanded into over 200 students with a shared interest in science and technology. Most STWing members live together on special floors of two of the College Houses at Penn.

STWing has benefited from a close association with Penn's School of Engineering and Applied Science, the subject of a paper presented at the FIE '97 conference **[4]**.

We look first at the most formal attempt at Computer Science education attempted within STWing, and discuss the results of the program. We then look briefly at other, more informal programs within STWing.

Introductory Computer Science Curriculum

During the freshman year of the Computer Science and Engineering curriculum at the University of Pennsylvania, the only major-related coursework that students normally take is a two semester course titled "Programming Languages and Techniques," which includes introductory programming laboratory work. Later in this paper, the course will be referred to as CSE 120 and CSE 130, which are the first semester lecture and lab sections respectively, and CSE 121 and CSE 131, which are the second semester lecture and lab sections, respectively.

The lecture and lab are actually one integrated course and are graded together; for each semester, the lecture section is worth 1.0 credit units (1.0 credit unit is equivalent to 3 course hours per week) and the lab is worth 0.5 credit units. The lecture is taught by a tenured faculty member, and the labs are taught by Computer and Information Science graduate students as part of their teaching practicum.

During the 1997-1998 academic year, these courses covered Scheme, SML/NJ (Standard ML of New Jersey), and Java over the span of two semesters; each language was addressed for 7, 12, and 8 weeks, respectively. The course started with Scheme, a very simple language for teaching basic programming concepts, because of its loose typing and the lack of imperative statements (i.e., the only purpose of code was to accept an input value, and return the result of a calculation, albeit complex). Progressing into SML/NJ, the students were presented with a more rigorous language that was strongly typed, and were introduced to some concepts of "side-effects" caused by imperative statements.

About the middle of the second semester, the course switched from SML/NJ to Java, a conceptually and syntactically very different language. Java was not initially approached in an object-oriented manner, and little effort was made to compare the advanced code packaging features of SML/NJ to the class and object concepts of Java.

In an effort to provide students with a more holistic, object-oriented approach to Java, one of the student authors, and another STWing member, both of whom had previously taken the introductory computer science courses proposed to teach an additional section of the second-semester CSE 131 lab. This was possible primarily because all evaluation and grading is done through the lecture portion of the course, allowing the student teachers to focus on different and additional material.

STWing Supplementary Laboratory

New Paradigm

We planned the supplemental lab section with the ultimate goal of utilizing an object-oriented curriculum and a faster overall pace. The idea was to provide the students with a more comprehensive and conceptually sound understanding of the Java language than could be provided through the standard curriculum of the course. Because the initial weeks of the course still dealt with SML/NJ, we planned to take a double-headed approach, where we would provide any relevant information necessary for SML/NJ assignments and examinations, while at the same time teaching the fundamentals of Java. Because we would start Java so early in the semester, we would have a jumpstart on the Java instruction in the course lectures.

Even though the STWing lab and the normal course lectures and labs would not be handling similar material contemporaneously, the curriculum for our lab section was designed to work hand in hand with the lecture material. We focused on basic concepts and effective programming, while the lectures would fill in details regarding syntax and other properties of the Java language. We would also be able to further demonstrate the logical progression between languages in the course (from Scheme to SML/NJ to Java) and how they were in many ways very similar.

In addition to those expected from the modified curriculum, we also saw important benefits from the fact that the teachers of the lab section would be peers, both *academically and residentially*, with many of the students.

Implementation

STWing presented this supplementary lab section in the Spring of 1998, to 12 students of the CSE 121/131 course. The lab was run not as a replacement for the existing lab and lecture sections, but rather as a voluntary addition to the curriculum.

Leading the section were two sophomores who had taken the course the year before, Walter Rice and Jon Kaplan. Both had mastered the material of the course, and Jon in particular was very proficient with Java having worked with it extensively as an intern at Sun Microsystems.

The section was presented two nights a week, and utilized the standard undergraduate computer laboratory classroom used for all introductory lab sections.

In general, the STWing lab section proceeded somewhat behind the rough schedule that had been planned, primarily because the teachers discovered that the students required more assistance with the lecture material than had been anticipated. Also, the double-headed approach at teaching SML/NJ and Java simultaneously was very time-inefficient, in that the half-hour provided for each was not enough to complete a useful exercise.

To help counteract the time limitations, some of the additional material was presented asynchronously; practice problems and peer presented "lectures" were posted on the web, or sent to the students via the section's e-mail list server. The material prepared for this web page serves as a general intro to the fundamentals of Java, and continues to be used as a reference by the former students.

Discussion of Results

The primary benefits of the STWing lab section were increased course flexibility and greater personal attention, both of which were possible due to a significantly lower student to teacher ratio than in normal lab sections. This flexibility ameliorated a situation that otherwise would have been greatly detrimental to the success of the section.

In addition, because the peer instructors had previously taken the same introductory course, and had advanced through more of the Computer Science and Engineering curriculum, they had a good understanding from a student's perspective of the knowledge required for further coursework. They also were able to draw upon their experiences outside the academic setting and introduce real world considerations that are sometimes overlooked in an academic programming course.

Peers of the students, members of STWing, and the instructors, who lived on the same floors as many of the students, were available at many times other than those set aside for the lab section.

The instructors also benefited from teaching the lab section, particularly through the need to prepare material from scratch and present it to a varied audience, in terms of knowledge and learning speed. Understanding that only two instructors is a very small universe to draw conclusions from, we still recognize the skills and experience gained from teaching the section.

However, although the STWing lab section had noticeable benefits, it suffered from some problems. For instance, the particular group of students who participated in this lab section tended to need more review of material taught in the lectures than was expected. Although the situation was beneficial to the students in that the material was flexible enough to address these unforeseen needs, the course was not able to progress into many of the advanced topics that had been planned.

Students also indicated confusion over the presentation of different material at different times in the lectures and the labs, as well as confusion caused by the two approaches to Java (object-oriented v. non-object-oriented).

Analysis

The problem that we see with the supplementary lab section is that it was faced with trying to serve two very different audiences: students who desired an accelerated curriculum, and students who needed additional review and practice with the normal material. This tended to frustrate the efforts of the lab instructors, either to present material faster, or to stay with material that needed repetition. At either extreme, one group of students (the lab was divided almost equally between the two types) would have gained nothing from the supplementary lab.

In addition, the supplementary lab cannot be separate in content from the main curriculum because of the confusion caused, even in those students able to learn at a faster pace.

The two very disparate needs that we have recognized cannot be served in the same manner. Students desiring a faster pace seem to also require that the entire course move at that pace. It isn't that the material in either the lecture or the accelerated lab is difficult, but rather the coordination of the two causes confusion and other difficulties. As a student group, we cannot address this need for an entire accelerated curriculum directly; however, we can look forward to addressing the needs of the other students.

The idea is to provide a creatively repetitive, contemporaneous supplement to the course material to enhance understanding, and not seek to present an increased base of knowledge. The supplementary section would have been more successful with regards to students requiring review of the lecture material if it had been more in the form of the Supplemental Instruction (SI) program described by Webster and Dee in "Supplemental Instruction Benefits Students in an Introductory Engineering Course" [5]. The SI program described in that paper utilized peer, student "facilitators" in an active learning environment.

The major differences between the SI program and the situation in STWing is that the effort in STWing was undertaken entirely by students; we still strongly feel that this is an important ingredient in a successful peer education program. Particularly for the instructors, there is an enormous benefit to having the responsibility for curriculum design and implementation; through similar programs, students can learn presentation and teaching skills that are not readily available elsewhere in the University.

Mini-Courses

On a more informal basis, students within STWing have started to organize mini-courses that teach various computing or programming skills necessary for future course-work, personal research, or technological competency.

The first such course to be offered was Teach-HTML, which taught the basics of HTML, JavaScript, and CGI programming in four Saturday sessions. Successfully presented to 15 people in Spring 1999, this course was offered to residents of Kings Court/English College House, the home of the STWing freshmen residential program. The course was presented by STWing freshmen who had experience in the subject area; however, none of the freshmen teachers had ever taught before a group before.

We have seen in Teach-HTML many of the benefits of the STWing introductory lab section, without many of the drawbacks. Because the courses are peer-lead, the students gain the benefit of increased instructor availability and curriculum flexibility, and the teachers are learning to teach to a group that has very differing interests and skill sets.

Through the presentation of very short, topical courses, the problems of confusion, in terms of non-contemporaneous presentations of the same material in different ways are avoided. Since the course is not about graded material, the atmosphere is more relaxed, and students are free to learn at their own pace. Attendance was greatly improved because there was nothing other than a desire to learn that motivated students to come in the first place.

Over the course of the Spring 1999 and Fall 1999 semesters, STWing members will be presenting additional mini-courses, in the areas of C and C++, Perl, and basic electrical engineering for non-majors.

In addition, STWing graduates now active in professional fields have expressed interest in returning to present multiple-session, hands-on mini-courses, such as an introduction to robotics and embedded programming.

The Peer Environment

So far we have looked at the very formal, organized implementation of a supplementary lab section in cooperation with the standard computer science curriculum, as well as independent efforts to present computer science topics to a wider audience, primarily within the residential system at Penn. We now mention briefly the environment with the STWing program that fosters peer cooperation and learning, particularly within the confines of computer science.

Students live and work in an environment in which resources, both in terms of knowledge and equipment, are readily available. STWing maintains its own computing facilities, and provides students with computing resources that cannot be provided by the engineering school. The students also have access to a resident systems administrator, again an undergraduate. The systems administrator has been a key to the success of the program.

Future Plans

Recognizing the benefits that exist from being able to effectively present technical material to others, STWing is proposing a program through which participants in the peer education programs of STWing would receive academic credit for their work. This credit would also be granted for participation in undergraduate research projects, and is offered as an incentive to participate in research and peer education while an undergraduate.

As part of our interest in science and technology, STWing will also be developing a "minimum basic skills" set of essential requirements for computer literacy in the 21st century. Through presentation of mini-courses and supplemental instruction in the residences, we hope to provide students, especially non-computer science majors, with increased technology competency.

References

[1] Light, Richard J., "Explorations with Students and Faculty about Teaching, Learning, and Student Life." *The Harvard Assessment Seminars, Second Report.* (1992)

[2] Love, Patrick G., "Exploring the Impact of Student Affairs Professionals on Students Outcomes." *Journal of College Student Development*, Vol. 36, No 2, pp. 163-170. (1995)

[3] Lumpe, Andrew T., "Peer Interaction in Science Concept Development and Problem Solving." *School Science and Mathematics*, Vol. 95, No 6, pp. 302-309. (1995)

[4] Santiago-Aviles, Jorge J., M. Krimo Bokreta, and John C. Parker, "The Partnership Between the Science and Technology Wing and the School of Engineering and Applied Science: An Experiment in Living and Learning," 1997 ASEE/IEEE Frontiers in Education Conference Proceedings. (1997)

[5] Webster, Thomas, and Kay C. Dee, "Supplemental Instruction Benefits Students in an Introductory Engineering Course," 1997 ASEE/IEEE Frontiers in Education Conference Proceedings. (1997)