Peer-Centered Service Learning

Mark A. Holliday [1] and David R. Luginbuhl [2]

Abstract - **We believe that improved learning by the undergraduate computer science major is intrinsically linked with improving student engagement. We have developed a framework aimed at encouraging student engagement. Most of the individual ideas are not especially innovative. However, we think others will be interested in how we have integrated them and made them a visible and central part of our undergraduate educational experience. We focus on three particular areas identified as critical for improving student engagement: student-faculty interaction, active and collaborative learning, and experiences outside the classroom. It requires that students take on roles that either previously did not exist or were the sole province of the faculty. The key to this approach is the integration of two techniques that others have identified: peer learning and service learning. Because the service learning is centered on the peer of the student providing the service we use the term peer-centered service learning.**

Index Terms - Collaborative Learning, Peer Tutoring, Service Learning, Student Employment.

INTRODUCTION

While the computer science education community has been exploring how to improve student learning for computer science majors, parallel activities are ongoing in the broader context of improving undergraduate education across all disciplines. In both cases, it has become increasingly clear that there are some common traits and techniques that are effective in improving learning outcomes. The most succinct characterization of these features is that an engaged student is likely to be a successful student. At Western Carolina University, we have considered how we can create an environment that encourages such student engagement. We believe that the explicit and conscious decision of computer science faculty to create and maintain such an environment is a key step towards the goal of a positive educational experience for undergraduate students.

We have developed a framework we call peer-centered service learning that creates such a learning environment. This framework has evolved by incorporating ideas both from the computer science education literature as well as the broader undergraduate education literature. Most of the individual ideas are not especially innovative. However, we think others will be interested in how we have integrated them and made them a visible and central part of our undergraduate educational experience.

The National Survey of Student Engagement (NSSE) [6] has been conducted for several years. It measures student engagement (through student surveys) in five categories: (1) academic challenge, (2) student-faculty interaction, (3) active and collaborative learning, (4) enriching educational experiences (beyond the regular experience of being a student in a classroom), and (5) supportive campus environment. Of relevance here are the middle three categories: student-faculty interaction, active and collaborative learning, and experiences outside the classroom.

Our approach to addressing these three areas requires that students take on roles that either previously did not exist or were the sole province of the faculty. The key to this relationship is that we integrate two techniques that others have identified: peer learning and service learning. Because the service learning is centered on the peer of the student providing the service we use the term peer-centered service learning.

In the next section we identify the influences in the literature on our approach and the detailed goals we are trying to achieve with respect to the three middle NSSE categories. We then present the specifics of our approach. Following that, we cover the mechanism that we use to implement our approach. We end with a short summary in the last section.

INFLUENCES AND DETAILED GOALS

As mentioned in Section 1, the National Survey of Student Engagement has played a role in our conclusion that improved learning by the undergraduate computer science major is intrinsically linked with improving student engagement and that improved student engagement requires increased student-faculty interaction, active and collaborative learning, and enriching educational experiences beyond the standard classroom role as a student.

A second important influence is service learning. Service learning has been recognized as an effective technique for enhancing the educational experience of students. Typically, service learning means learning through community service projects that have been designed to fit within the curriculum and often as part of a particular course [3,9]. We assert that providing service to other students when done within a well designed framework can serve as an important complementary form of service learning that leads to many of the same benefits.

Further motivation for peer-centered service learning can be found in the labor programs at some colleges such as Berea College [1] and Warren Wilson College [10]. In these colleges, an important part of the educational experience is the

[1] Mark A. Holliday, Department of Mathematics and Computer Science, Western Carolina University, holliday@email.wcu.edu
[2] David R. Luginbuhl, Department of Mathematics and Computer Science, Western Carolina University, luginbuhl@email.wcu.edu

work component. Every student must work within the labor program for ten or so hours each week. These jobs are not typical student work-study assignments; they are instead the type of jobs one would find in campus facilities management (e.g., locksmith and carpenter). Some of the benefits of a labor program are that it is a form of experiential learning, it develops a work ethic, and it provides practice in working with others.

We have also been influenced by studies that show the impact of teaching techniques such as active learning, collaborative learning, and peer learning [4,11]. Some researchers have shown that including these techniques within the classroom can improve student learning outcomes as measured by the percentage of students in CS1 that finish with a grade of D, a grade of F, or withdraw [2].

These influences led us to develop the following specific goals.

1. Our introductory courses (our CS0, CS1, and CS2) need to have a small class size, with much of the class time spent in exercises involving the traits of active learning, collaborative learning, and peer learning.
2. We want to involve as many of our computer science majors as possible in service learning by helping other students in computer science courses.

These two goals are interdependent: even with a small class size (say, twenty-five students), it is difficult to achieve the desired level of student-faculty interaction for active learning with just one instructor. By having more experienced computer science students helping in the classroom, we are able to achieve the desired amount of student-faculty interaction and engage the more seasoned students in service learning.

It is the intent of this paper to encourage a rethinking of how undergraduate students can be involved in helping to provide a high-quality learning experience within an undergraduate computer science program while gaining from the experience as well. Structured properly, such involvement becomes a form of service learning that is centered on passing knowledge to other students.

In peer-centered service learning, computer science faculty members intentionally design opportunities for undergraduate students to apply what they have learned in the computer science curriculum as they assist their peers. Integrating the concept of service learning with a focus on other students and on course content is a key technique in providing a high-quality educational experience for undergraduate computer science majors. This experience draws strengths from three areas:

1. *Service learning*: As noted in the literature [3], any valid form of service learning is beneficial because the students receive positive feelings from helping others. Furthermore, students gain valuable teamwork skills both as members of service provider teams and by interacting with those who are receiving the service. We are careful to structure our activities to ensure that these benefits also occur.
2. *Reinforcement of learning*: Students apply their knowledge of computer science in teaching other students or in maintaining the software and hardware facilities of the computer science program. As the saying goes, "the best way to really learn a subject is to teach it." We adhere to that philosophy by designing most of these service learning opportunities so that they involve teaching other students. In fact one of the questions in the NSSE survey addresses how much time students have spent tutoring others.
3. *Interactive learning*: The quality of the learning experience for the students taking our lower-level courses is greatly enhanced by the involvement of upper-level students. In any course, but especially in introductory programming courses, we feel that students learn best if they have an opportunity to spend class time working on a project themselves individually or in a small group while having frequent interactions with an instructor or student helper. Having the upper-level students there to assist the students in the classroom is essential in ensuring that enough attention is given to each student in the class.

THE SPECIFICS

As discussed earlier, by peer-centered service learning we mean a formal mechanism through which we provide opportunities for undergraduate computer science students to work with other students learning computer science within the framework of the activities of our computer science program. For the purposes of discussion, we have organized these activities into four categories, outlined in Table 1. Note that two of the four main categories involve volunteer (unpaid) activities, while the other two involve paid positions.

TABLE I
FOUR CATEGORIES OF PEER-CENTERED SERVICE LEARNING

Category Type	Category	Activities
Unpaid	Groupwork	Informal Leader
	High School Programming Contest	Instructor Proctor Grader
Paid	Course Content	In-Class Assistant Lab Assistant Evening Tutor
	Departmental Computer Systems	System Administrator Lab Manager Webmaster

The first category is the most subtle in that sometimes the students involved do not even realize that they are participating! It involves our CS1 and CS2 courses, which are both taught using Java. In these two courses, which we limit to twenty-five students per section, we incorporate a significant amount of in-class groupwork. In these courses, groupwork involves having the students divide themselves into groups of three or four. Each group is placed at a different area on one of the blackboards (the room needs considerable blackboard space) and works on writing a small program or

program fragment to solve a problem posed by the instructor. When a specific group believes it has solved the problem, it is required to describe or demonstrate that solution. As the instructor observes the dynamics of the groups he may move the stronger students around so that each group has at least one leader. This is done informally; no student is designated as a leader. In fact over the semester, this leadership role may change.

Because a group's demonstration is completed only when all the members of the group can explain how the program works, the group leader is motivated to help the other students learn. By designing the classroom experience so that students in the class can assume such leadership roles, we have created a form of peer-centered service learning. Although this role is less formal than the roles in the remaining categories, the students involved in the other three categories often started as groupwork leaders in our CS1.

The second category involves volunteering for the programming contest we host for high schools in our region every spring. We rely on our computer science majors' assistance with various phases of the contest to ensure its success. This includes teaching the high school students how to use the contest programming environment, proctoring the contest, and testing submitted solutions.

The last two categories are paid positions. Category three involves course-specific student interaction and has two subcategories. One subcategory is that each section of our CS0, our CS1, and our CS2 has an upper-level computer science major as a class assistant; in addition, our CS1 and CS2 each have an assistant in weekly laboratory. The class assistants work with the students in the classroom or laboratory one-on-one or in small groups.

Our CS0 provides a gentle introduction to programming for students who are not planning to major in computer science. We use HTML and Javascript. All the class time is spent in an electronic classroom with no more than twenty-five students per class. After a short lecture by the instructor, students work on assignments individually. The class assistant spends his or her time going from student to student answering questions and having the students do demonstrations when they complete an assignment.

As discussed earlier our CS1 and CS2 use groupwork extensively when meeting in the regular classroom. Since there could be as many as seven groups in a class it is difficult for the instructor to be able to spend enough time with each group. The class assistant also works with each group answering questions, having the group demonstrate completion of a program, and asking questions of the group members to make sure everyone understands the concepts.

The lab assistant also performs a similar role during the weekly closed lab of the CS1 and CS2. The difference is that the closed labs are in an electronic classroom (so the students are each using a computer) and the students are working individually or in teams of two. In this respect, the CS1 and CS2 lab assistant functions in much the same way as the CS0 class assistant.

It is important to clarify what class and lab assistants are not; they are not student graders. The central purpose of the student assistants is to help and to interact with students. The peer and helping relationship between the student assistant and the student is inevitably changed and compromised if the student assistant is also assigning grades. Furthermore, we are concerned that the idea of a fellow undergraduate assigning grades adds a level of discomfort for both the student assistant and the student being assisted.

On the other hand, we do want to have the student assistant help the student and the instructor by verifying when the student has completed some kinds of assignments. Our solution has several components. First, the student assistants do verify completion of the in-class groupwork. However, that groupwork does not count towards the course grade. Second, the only graded work that student assistants verify is the CS0 assignments and the labs for CS1 and CS2. They do not verify programming projects or help grade tests. Third, they only verify that students have completed assignments in that they are only permitted to provide a binary determination (completed or not completed). If there is any partial credit to be given, only the instructor may make that determination. Fourth, any student always has the option of requesting that the checking be done by the instructor instead of by the student assistant. Finally, if the student assistant determines that the student has not completed the assignment, the student can continue work on the assignment and then ask to be rechecked by either the student assistant or the instructor. Rechecking can continue until the next test in CS0 or the end of the lab session in CS1 and CS2.

Another activity associated with course content involves tutoring. On most weeknights, we have student tutors available in our departmental computing laboratory. They work one-on-one with students in our CS0, CS1, and CS2 courses.

The second paid category involves upkeep of our department's computer systems. This category originally emerged out of necessity. Our university provides extensive university-wide computing facilities but provides no support for departmental facilities. As in any computer science program, we have some special needs, so we run several servers of our own, as well as a small general purpose (ten machines) computing laboratory. Our servers run the Linux operating system and host our departmental website along with providing user accounts and several other services. Our client machines are a mixture of Windows and Linux machines. All of these facilities are maintained by undergraduate computer science majors under the supervision of one faculty member.

The system assistant positions enhance the students' learning experience. The students put into practice many of the concepts that they have learned in their computer organization, operating systems, database systems, and computer networking courses. They have had to do a significant amount of programming both in the form of Perl scripts and custom applications. Potential employers are attracted by the work experience. There is also a significant

degree of student interaction. The system assistants are the primary people who answer questions from students about how to use the departmental facilities and how to resolve problems. As with all the service learning opportunities, we emphasize the importance of treating the students that are being helped with courtesy and respect.

MECHANISM

Because we consider peer-centered service learning to be a key component of the educational experience of our computer science majors, we are interested in maximizing the number of students who participate. Requiring participation is not feasible, but we make every effort to encourage participation. Due to space considerations, we briefly describe only our approach to paid positions.

Near the end of each semester, after the students have pre-registered for the coming semester (so they know when they will be available) a web-based application that allows students to apply for paid positions is posted and announced. Applicants identify their job preferences, when they are available, specific indicators of academic progress (such as GPA), and any related experience.

One faculty member has the responsibility for organizing this information into a format that facilitates matching students to jobs. The computer science faculty members meet to assign students to particular positions.

We attempt to assign students to positions in which they will perform well, but we also try to distribute the assignments over as many of the students as possible. For example, during the last academic year we had 38 upper-level computer science majors; 13 of them were in paid service-learning positions within the computer science program during the Fall 2003 semester. To maximize the number of students involved, the number of hours per week assigned to each student is relatively small (3-5 hours).

Finally, the faculty member in charge of assignments contacts the students to offer the positions and supervise any revisions due to schedule changes. Our experience has been that having a single faculty member assigned to preprocess the applications before the meeting is important. Otherwise the amount of cross-referencing that has to be done during the meeting becomes too time-consuming.

IMPACT, BENEFITS, AND CONCERNS

As described above, we had 13 computer science majors in paid positions during the Fall semester. Around ten upper-level computer science majors helped with our high school programming contest last year. Even accounting for overlap between students in these two groups, about half of our upper-level majors participate in the service learning component of our program. If we include those upper-level majors who had been groupwork leaders in CS1 or CS2, the participation rate would be even higher. Thus, we feel that we are achieving our goal of maximizing peer-centered service learning opportunities as a means of increasing student engagement.

We have been using this approach for eight years, and the structure of it has been stable for a number of years. From our experience, we can identify a number of benefits both to the student in the position as well as to the program as a whole. The benefits for the student include:

1. reinforcement of computer science concepts through teaching others and managing departmental computing facilities.
2. improved communication and interpersonal skills
3. increased confidence in their own abilities as computer scientists
4. opportunity to interact with faculty in a context other than the usual teacher-student setting
5. positive attitude emanating from the pleasure of seeing others helped by one's efforts
6. enhanced employment prospects.

The benefit of enhanced employment prospects occurs in at least two ways. First, students are able to list real job experience on their resumes. Second, and perhaps more important, these positions provide students with skills that employers are looking for but that are often missing from an undergraduate curriculum, according to the National Association of Colleges and Employers (NACE) [5]; namely, communication, teamwork, and interpersonal skills. In fact, peer-centered service learning addresses directly several suggestions found on the NACE website:

- "Develop your interpersonal and teamwork skills...."
- "Get some work experience while you're in school…"
- "Learn what the workplace is really like." [5]

The benefits for the program include:

1. incorporation of active learning into the introductory computer science courses. We believe that the learning experience is much greater in a class where each student is an active participant who gets to apply his or her understanding, receiving feedback interactively during the class time. No institution can afford to hire enough instructors to make this possible. Class assistants and groupwork leaders make this possible and in some ways are preferable to more instructors (since the other students can relate more easily to them as peers and since there is no tension because the student helpers are not involved in assigning grades).
2. ability to offer a tutoring program for students in the introductory courses.
3. outreach to the region through the programming contest for high schools.
4. increased retention of computer science majors. This includes the upper-level students providing the services and the lower-level students who consume the services.
5. recruitment, since we can advertise improved skills, abilities, and employment record of students graduating from our program. We always mention our peer-centered service learning opportunities during presentations at our High School Open Houses.
6. ability to support a departmental computing facility at relatively low expense.

7. continuity for new faculty and for faculty new to a particular course. This has been an unexpected but important benefit. New faculty members are sometimes concerned since they may be assigned an introductory course that they have not taught for quite some time or that uses a language (Javascript, Java) that they have not used or that employs an approach they have not used (objects-early, extensive in-class groupwork).

The new faculty member does receive copies of the course materials from the faculty who have already taught the course and can ask questions of the other faculty. However, the class assistant has turned out to be a great additional resource. The class assistant is in the classroom with the instructor during all the class times so the class assistant knows what the new instructor is encountering. The class assistant also knows how the course has been taught in the past. Consequently, the instructor can ask the class assistant for advice about how to present some material and how to deal with problems that arise in the classroom.

A computer science program considering developing a formal peer-centered service learning component needs to address some possible concerns.

1. There is some cost for the paid positions. However, we would argue that this is one of the most cost-effective ways to improve student retention and learning. The cost is surprisingly low since each student is only working for a few hours a week and the wage rate is the university-wide student wage rate. Our university provides funds for tutoring, and we are able to use these same funds for our paid student positions. An alternative approach is to charge a lab fee as part of the cost of the lab courses (CS0, CS1, and CS2).
2. A faculty member must spend some time each semester coordinating the job applications, meeting, and subsequent contact with the students.
3. A faculty member must supervise the system assistants who manage the computing facilities.
4. Faculty members may have to make clear that class assistants are not graders. They are there to help the students through one-on-one and small group interaction.
5. Incorporating peer-centered service learning implies that the program must commit to introductory classes that are interactive and small (as noted above, our class size for CS0, CS1, and CS2 is around twenty five students). A program that teaches these courses as large non-interactive lectures would have to make some fundamental changes. Small enough introductory classes are possible even at a state university (such as ours), but only if there is institutional support for small introductory classes and faculty willing to teach the corresponding number of contact hours.

The need for small introductory classes in the proposed approach contrasts with an alternative model of undergraduate involvement in the computer science program that has developed at some larger research universities [7, 8]. In that model experienced undergraduate computer science majors become undergraduate teaching assistants for the introductory programming course. The course is taught as a large lecture with the undergraduate teaching assistant leading a section of the class in an hour-long session each week, as well as grading [7].

We have concerns about the appropriateness of giving such a level of authority to the undergraduate teaching assistant in both leading a section and in grading the work of fellow undergraduates. In addition, the large lecture appears to be a significant part of the undergraduate teaching assistant model. We feel that small class size for all parts of the course is important because it allows active and collaborative learning and thus improves the learning process. Another distinction is the emphasis of our approach on maximizing student engagement. For the paid positions we intentionally have each student work only for a few hours a week instead of ten hours a week [7]. The reason is so that we can distribute the work among as many students as possible. This increases the complexity of supervising the students, but it is the best choice given our goal of maximizing the number of students involved. Finally, our approach views paid positions within the classroom as only one option within the broader framework of peer-centered service learning. In other words, the approach we propose includes techniques for involving a large fraction of the undergraduate computer science students at all levels of experience with levels of leadership that are consistent with their being undergraduates.

SUMMARY

Our key goal is maximizing the engagement of our students in their experience as computer science majors. With peer-centered service learning, we have developed a framework for involving the largest number of students possible. At least some of the activities we have identified are likely to be going on in any undergraduate computer science program. Our key point is that often, the potential for these activities is neither appreciated nor exploited as it should be. The benefits to the students themselves, to their peers, and to the faculty are significant, and the cost can be low.

Recognizing that these activities are in fact a form of service learning is critical. The same benefits that arise from traditional service learning can also come from these types of peer-centered service learning. The key is to make these activities a formal part of the educational experience of your program.

REFERENCES

[1] Berea College Catalog 2003-2004, Student Labor Program, Berea, Kentucky, http://www.berea.edu/stlife/labor/default.html.

[2] Chase, J.D., and Okie, E.G., "Combining Cooperative Learning and Peer Instruction in Introductory Computer Science", *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, 2000, pp. 372-376.

[3] Barbara Jacoby and Associates, *Service Learning in Higher Education: Concepts and Practices*, Jossey-Bass, 1996.

[4] Miller, A. and Kay, J., "A Mentor Program in CS1, *Proceedings ITiCSE*, 2002, pp. 9-13.

[5] National Association of Colleges and Employers, What Employers want vs. what they see in job candidates, http://www.jobweb.com/joboutlook/outlook2.htm/.

[6] National Survey of Student Engagement, 2001 NSSE Viewpoint: Improving the College Experience: Using Effective Educational Practices. Bloomington, IN: Indiana University Center for Postsecondary Research and Planning. http://www.indiana.edu/~nsse/.

[7] Reges S., "Using Undergraduates as Teaching Assistants at a State University," *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, 2003, pp. 103-107.

[8] Reges S., "The Effective Use of Undergraduates to Staff Large Introductory CS Courses," *SIGCSE Bulletin*, February 1988.

[9] Sanderson, P., and Vollmar, K., "A Primer for Applying Service Learning to Computer Science", *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, 2000, pp. 222-226.

[10] Warren Wilson College Catalog 2003-2004, Work Program, Swannanoa, NC, http://www.warren-wilson.edu/work.

[11] Wills, C.E., Finkel, D., Gennert, M.A., Ward, M.O., "Peer Learning in an Introductory Computer Science Course", *Proceedings of the 25th SIGCSE Technical Symposium on Computer Science Education*, 1994, pp. 309-31.