In *Proceedings of the ASEE/IEEE Frontiers in Education Conference*, Pittsburgh, PA, November 1997.

Study of a Group Project Model in Computer Science

Craig E. Wills, David Finkel
{cew,dfinkel}@cs.wpi.edu

Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609

Abstract

We have explored the use of peer learning in a large, introductory data structures course within our computer science curriculum. The principal peer learning activities are group programming projects where each student group is assisted by a trained upper-class undergraduate peer learning assistant (PLA). The PLAs are consultants to the group to help facilitate group interaction and support group and individual questions.

This paper describes the use of group programming projects in the course and studies their use in three instances of the course over a four year period. Each of these courses was taught by the same professor during the same time in each academic year. Each course was taught with the same group programming project model, but with variations in the number of students, how the groups were organized and the number of PLAs that were used.

Overall, we feel the introduction of group programming projects in this large introductory course has been beneficial to students, PLAs, TAs and faculty. The students are able to legitimately work with peers to solve problems and learn the material. Peers are more able to provide a student perspective in answering questions. The results shown in the paper indicate the long-term viability of such an approach. The approach has continued to rate well with students through variations in how it is used.

1 Introduction

A traditional classroom environment utilizing lectures and assignments done individually by students is an efficient medium for the transmission of factual knowledge, but is poorly suited to developing higher-level cognitive skills and increasing student motivation. This environment makes it easy for students to become isolated in courses and does not take advantage of peers working together.

In contrast, peer or cooperative learning has students working together as part of their own learning experience. Peer learning can take many forms. Although there has been much group work at the upper levels of the computer science curriculum, we believe that the large class sizes often deter its use in introductory courses. However, these courses are where this technique can be most useful. We have explored the use of peer learning in such a course—the introductory data structures course (CS2) in our introductory curriculum.

In our work, informal group activities are done in class using group quizzes, but the principal peer learning activities are group programming projects. Each student group is assisted by a trained upper-class undergraduate peer learning assistant (PLA). The PLAs are consultants to the group to help facilitate group interaction and support group and individual questions.

Through the use of peer learning assistants and the creation of software to minimize the administrative overhead of groups we have been able to concentrate on making the course a quality educational experience. This paper describes the use of group programming projects in the course and studies their use in three instances of the course over a four year period. Each of these courses was taught by the same professor during the same time in each academic year. Each course was taught with the same group programming project model, but with variations in the number of students, how the groups were organized and the number of PLAs that were used.

2 Background

Group work has been done in many forms within different courses across the computer science curriculum [1, 2]. Other work has examined the use of undergraduate teaching assistants [3]. We have used group programming projects in a large data structures course within our computer science curriculum. This course is taken after the students have a solid background in computer programming. Our approach for using peer-based learning in the course has been described in [4]. The following provides background on the approach before moving on to a discussion on the evolution of the group programming project model in the course.

Group projects are done outside of class and focus on larger design and programming problems. There are three such assignments during a course following an initial assignment done on an individual basis that leads into the group work. The projects are done in pre-assigned groups of 4-5 students. Each group is led by a PLA to help facilitate group activity. All students on the project receive the same grade unless individual assessments indicate otherwise.

Figure 1 summarizes the lifecycle of a group project and shows the roles of the instructor, teaching assistant (TA) and PLA. The instructor of the course is the overall manager for the activities in the course. He or she creates the group assignment and is ultimately responsible for questions or problems. As shown, the design for the project is done by the entire group and broken into parts for individual work. Once individual pieces are coded and tested, the parts are built together and tested. The PLA helps facilitate group work and is available for technical questions during all phases of the project.

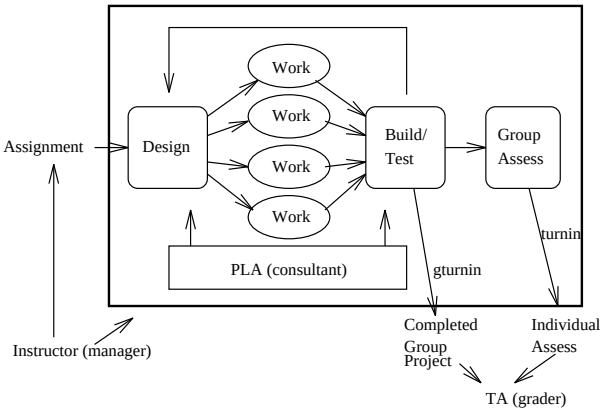

Figure 1: Project Lifecycle for a Group Project

Details of the assignment are covered in a peer group meeting outside of class and laboratory time. If possible, this meeting includes the PLA. Students discuss the overall organization of the project and how it can be divided for work among the group members. With the help of the PLA, students take responsibility for pieces of the project based on individual styles and interests.

After distributing work among individual group members, the students work on their portions and communicate (often electronically) with each other and their PLA on progress and problems. Group work is facilitated with an electronic mail alias for each group and a group-specific directory where group members can do their work and share it with the group. The students have access to each other, the PLA, TAs and faculty for individual or group assistance. The students' work culminates by building the individual modules together into a working whole and testing them on various data. Students turn in their group contributions electronically (using *gturnin*) along with an individual evaluation of each group member's contribution to the whole (using *turnin*).

3 Variations on Our Group Project Model

Three offerings of the course using this basic model have been taught by the same faculty member at the same time during the academic year (both authors have used this model in the course with this paper focusing on the common experiences of one author). Each of these courses has its own variations, but the basic model has been constant. Variations in the three courses are summarized in Table 1.

Table 1: Offerings of Course

Year	Class Size	# PLAs	Group Org.	Lang.
1993	128	6	random	C
1994	130	2	self-assess	C
1996	187	6	self-assess	C++

The initial offering of the course was in 1993 with 128 students. Six PLAs were selected from student applications and these PLAs were trained for their role in the course (PLAs are paid for their time devoted to the course). Students in the class were organized into groups based first on the lab section they attended (labs contain 20-22 students). In this way

students were assured of finding their group members during lab time. In addition, students were surveyed on their out-of-class availability for meetings and organized into groups of 4-5 accordingly. Other than lab and out-of-class time availability, the students were grouped randomly.

One result of our initial offering was that the randomly formed groups left a few groups lacking interpersonal or technical leadership. These groups often had problems in successfully working on projects, which led to group conflict and often a visit to the faculty member's office by one or more group members. This result led us to reexamine how groups were formed. We considered grouping students by year or major, but we felt that diversity was important. Some students suggested choosing their own groups, but in such a large class with students not knowing each other well this approach did not appear easy to implement nor equitable to all students.

Based on our observation that successful groups had technical and interpersonal leadership, we modified the way in which groups were formed in the 1994 class (using the same approach in 1996). The results of five survey questions (given as part of a survey at the beginning of the class) were used to compute a "goodness" factor for each student based on the student's own self-assessment. Three questions are averaged to obtain a "group skills" rating while two questions determine a technical rating. In computing the overall goodness factor for a student, the technical rating is counted twice the group skills rating. Once each student's goodness factor is determined then groups were again formed based on a common lab time and where possible common out-of-class meeting times. However, an overriding concern was to equalize the average goodness rating for each group.

Part of the motivation for forming groups that worked well together in 1994 was that we employed only two PLAs. These PLAs were experienced from 1993, but with only two of them, there was less personal contact and much more use of electronic communication with the respective groups. We took this approach in 1994 because we had experienced PLAs that would not need training and it was an experiment on our part to see if the number of PLAs could be reduced. We thought this reduction had merit based on our observation in 1993 (and in subsequent years) that student groups weaned themselves from their PLA over time. In many cases, student groups would have little contact with their PLAs in either personal or electronic form, but work fine as a group. Only the groups with conflicts would continue to involve their PLA.

In 1996,the course population increased dramatically to 187 students. This increase was not linked to the approach used in the course, but rather based on a large increase in majors and general interest in computing. We used our same model, but did increase the number of PLAs to six (although a higher student to PLA ratio than in 1993). Other than class size, the only other significant change was the use of the C++ programming language (no overloading or inheritance) in the course versus the previous use of C.

4 Results

We obtained information and feedback from the students in a number of ways: a survey at the beginning of the course (numeric responses), a survey at the end of the course (numeric responses), a survey at the end of the course (written responses) and a standard WPI course evaluation sheet.

All surveys of students (except the WPI course evaluation) were given online using a survey program developed for the course. As baseline information, Table 2 provides demographic information on the students in each course. As shown, the course is predominantly males. Gender was a factor for group formation in 1996 with females generally placed together in groups if the other group formation criteria could still be met. No formal analysis based on gender has yet been done. The largest group of students in the course are Freshman computer science majors, but there are large numbers of non-majors, particularly from the Electrical and Computer Engineering Department.

Table 2: Course Demographics

	1993	1994	1996
Gender (% Male)	90	91	89
Major (% CS)	43	59	53
Class (%Fr)	37	53	46

Quantitative student perceptions on aspects of each course were gathered based on a 1-4 numeric scale where one is strongly disagree and four is strongly agree with a given statement. Averages for questions selected from the final survey common to each course are shown in Table 3.

The results show few dramatic changes between the three versions, but overall the best results occurred in 1993. This result is predictable given that the 1993

Table 3: Average Course Survey Results(1=SD,2=D,3=A,4=SA)

	1993	1994	1996
I learned a lot in this course	3.4	3.3	3.2
I liked working on projects with other students.	2.9	2.9	2.7
The people in my group worked well together.	2.5	2.5	2.7
My PLA was very helpful to the group.	2.7	2.0	1.9
I enjoyed this class.	3.2	3.0	3.0
I worked hard in this class.	3.3	3.1	3.2
I feel I learned (1=much less, 2=less, 3=about the same, 4=more, 5=much more) because of working with others in group projects rather than working on individual projects.	3.3	3.2	3.0
I feel my course grade will be (1=much lower, 2=lower, 3=about the same, 4=higher, 5=much higher) because of working with others in group projects rather than working on individual projects.	2.9	2.9	2.7
How do you rate your overall level of satisfaction with this course? 1=very disappointed, 2=somewhat disappointed, 3=somewhat satisfied, 4=very satisfied.	3.1	2.9	3.1

version had the best student to PLA ratio along with the initial enthusiasm and determination to make a new approach work. The 1994 version went fine, but the drop-off in some of the indicators is likely the result of fewer PLAs and a little bit of a let down in instructor enthusiasm. The 1996 version was similar with the lower student to PLA ratio helping the ratings, but the large number of students hurting.

In terms of specific questions, there is a large drop off in the helpfulness of the PLA to the group in 1994 and 1996. The 1994 results are explainable based on using only two PLAs. The 1996 results indicate that the PLAs were not as visible to the groups as would be expected. Based on our observations, we would conclude that this group of PLAs was less aggressive in making themselves known to the groups than in the past.

Another interesting result from all three versions was the perceived amount learned versus course grade because of the group projects. In all cases, the amount learned was the same or higher than expected for individual projects, but on the other hand the perceived grade was always lower because of the projects. Our experience as faculty indicates the course grading was not changed because of the group projects. We believe the perception comes from better students who feel they are dragged down by having to work with less capable students in the class. However, it is precisely this interaction we are encouraging with groups where in the best case all students learn more.

As a final comparison of the three courses, we gathered information from the standard WPI course evaluation form given in all courses taught at WPI. Table 4 shows two results. First is the percentage of students who agreed or strongly agreed with fourteen specific questions on perceptions about the instructor of the course. This number is the most commonly used figure on campus to judge the "success" of a course from the student perspective. We also include the results of a single question on how much the students learned in the course. Overall the results are similar, with the 1994 version showing the worst results. All of the evaluations are considered excellent for our campus.

Table 4: WPI Course Evaluations (Pct. of A/SA)

	1993	1994	1996
14 Instructor Questions	95	92	94
I learned a lot in this course	93	88	89

In terms of less measurable results, we found that

in each successive version of the course there were fewer group problems that had to be dealt with by the faculty member. Most group conflicts were resolved by the group itself or with the help of the PLA. We attribute this trend to the improved grouping method, which has reduced the number of severe group problems. We have also improved on our design of the projects for easier decomposition into individual pieces of work within the overall group. This guidance has improved group organization for what each group member should be doing and how the pieces interrelate. The use of C++ has also helped with expressing the decomposition.

There is still a problem faced by groups of how to deal with weaker or less motivated students whose failure to meet group deadlines potentially impacts the success of the entire group. The strong interdependence of group programming projects can cause tension in the group when this situation occurs and is from our perspective consistently the most difficult aspect of using group programming projects. One potential solution is allow students to be "fired" from groups, but this may have the negative effect of removing a student from a group who needs help from others the most. We have not used this approach, but in rare instances have disbanded groups where it seems to be best for the students.

A final issue that was not specifically addressed in our survey of students, but of interest is the effect on socialization of students working in the dormitory rooms through campus network connections. In the past couple of years WPI has wired the dorms and there has been evidence from other areas of campus that this change has affected social interactions of students. The use of group projects may be less popular with students because it forces them away from an environment in which they like to work into public computing labs. However, in the long term this may be a positive effect of this approach. It is an aspect that merits more study in the future.

5 Summary

Overall, we feel the introduction of group programming projects in this large introductory course has been beneficial to students, PLAs, TAs and faculty. The students are able to legitimately work with peers to solve problems and learn the material. Peers are more able to provide a student perspective in answering questions. PLAs have benefited from the approach by involving themselves in the teaching process and having a much better understanding of group dynamics. The big benefit for TAs and faculty is to lighten the potentially burdensome number of questions and grading that arise with such a large class. We hesitate to think about handling such a large number of students without group programming projects.

The results shown in the paper indicate the long-term viability of such an approach. The approach has continued to rate well with students through variations in how it is used. We believe this result is significant as the success of many innovations can lessen as the initial enthusiasm for a change naturally dissipates to a steady state condition. This approach does take time to organize the groups and train the PLAs, but the time saved from handling student interactions is worth the effort. This benefit is in addition to the increased interactions among the students.

An outgrowth of our work has been sponsorship of an NSF-supported workshop on the application of peer learning to the introductory computer science curriculum [5]. This project has brought together a diversity of computer science educators for an ongoing process on the application of peer learning to the introductory computer science curriculum.

References

[1] J. C. Prey, "Cooperative learning in an undergraduate computer science curriculum," in *ASEE/IEEE Frontiers in Education '95*, November 1995.

[2] H. M. Walker, "Collaborative learning: A case study for CS1 at Grinnell College and UT-Austin," *SIGCSE Bulletin*, vol. 29, pp. 209–213, March 1997.

[3] E. Roberts, J. Lilly, and B. Rollins, "Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience," *SIGCSE Bulletin*, vol. 27, pp. 48–55, March 1995.

[4] C. E. Wills and D. Finkel, "Experience with peer learning in an introductory computer science course," *Computer Science Education*, vol. 5, no. 2, pp. 165–187, 1994.

[5] C. Wills, D. Cordes, D. Deremer, B. Klein, R. McCauley, and L. Null, "Application of peer learning to the introductory computer science curriculum," *SIGCSE Bulletin*, vol. 29, pp. 373–374, March 1997.