COLLABORATIVE PROGRAMMING IN AN ADVANCED PROGRAMMING WORKSHOP CONDUCTED IN A DISTANCE LEARNING ENVIRONMENT

Tamar Benaya, Ela Zur
The Open University of Israel
tamar,ela@cs.openu.ac.il

ABSTRACT

There is ongoing discussion regarding the benefits of collaborative programming in Software development. This paper presents the results of a collaborative programming research conducted at the Open University of Israel which is a distance learning institution. The research was conducted during the academic year of 2004/5 in the course "Advanced Programming in Java Workshop", given in the Computer Science (CS) Department. The workshop is based primarily on J2EE technologies and requires the students to hand in a final project. We investigated the collaborative programming work in the workshop which takes place in a distance learning environment. Our results support the use of collaborative programming as an effective programming technique.

1. INTRODUCTION

The Open University of Israel (OUI) is an institution of higher learning which offers a variety of undergraduate and graduate programs. The OUI has an open admission policy and is based primarily on distance education. There are many universities all over the world that offer distance education programs for off campus students in addition to their face to face mode of education. The combination of Universities which are both open and based primarily on distance education is not very common. A thorough description of such universities can be found in Guri-Rosenblit's book [7]. The open admission policy opens its doors to all those who wish to study towards a Bachelor's degree. Enrollment does not require matriculation or any other achievement exam or certificate from another educational institution. Though applicants are not required to provide proof of prior scholastic achievements, their academic achievements are the key to their success at the OUI.

The teaching methods practiced at the OUI combine two modes of distance education: traditional distance education based on written materials which are sent to the students at their homes and Web-based teaching. The Web-based teaching was introduced in the past decade because the increasing world-wide demand for distance education learning communities [6]. Both modes of distance education are based on many years of experience practiced at the OUI in the undergraduate and graduate programs [1, 2].

The distance education and self-study practiced at the OUI enables students located throughout the country, to pursue higher education without leaving their home towns. The method of study also provides conditions that meet the constraints of individuals who work, raise a family, manage a household or serve in the military. The method is not space or time dependent as it is not based on a central campus where lecturers and students gather. For example, in the last semester, 60% of the students in the course "Introduction to Computer Science" were located in the general vicinity of the OUI while 40% of the students were located in peripheral areas of the country.

The OUI offers several undergraduate Computer Science (CS) programs and a graduate CS program. The undergraduate CS programs include a B.A. degree in CS and several interdisciplinary B.A. degrees which combine CS with another discipline such as Economics, Management, Education, Psychology etc. A detailed description of the different undergraduate and graduate programs can be found in the OUI Website [11]. The requirements for the B.A. degree in CS include five courses in Mathematics and about 20 courses in CS. The interdisciplinary degrees include more basic Math courses and the rest of the courses are divided between CS and the other discipline.

As part of the undergraduate requirements in all CS programs, the students are required to take at least one workshop or seminar. One of the workshops offered is the "Advanced Programming in Java Workshop".

The next section describes the "Advanced Programming in Java Workshop". Following that, we describe a research which we have been conducting at the OUI regarding the collaborative work which takes place in the workshop.

2. WORKSHOP DESCRIPTION

The "Advanced Programming in Java Workshop" is an elective course which can be taken towards the end of the undergraduate degree. The prerequisites for the workshop include an advanced programming in Java course which must be completed with a grade of 70 or above, and an additional programming course beyond CS1 and CS2.

The purpose of the workshop is to expose the students to advanced state of the art technologies giving them up to date knowledge which will help them compete in the Hi-Tech industry.
The workshop is based primarily on J2EE technologies including: JavaBeans, Java Data Base Connectivity (JDBC), Servlets and Java Server Pages (JSP), Remote Method Invocation (RMI), Enterprise Java Beans (EJB), Java Messaging Service (JMS), Java2 Micro Edition (J2ME), JINI, JavaSpaces and XML.

The textbook used in the workshop is "Advanced Java 2 Platform - How to Program" by Deitel, Deitel, and Santry [5]. It covers most of the topics mentioned above. The students are required to give a lecture on one of the topics and to hand-in a final project which implements a least three of above technologies. Examples of some projects are: a Web-Based commercial system such as a bookstore or a department store, a Web-Based game server managing several games and players, a Web-Based ordering system such as cinema tickets, DVD rentals, restaurant take-outs etc.

The methodology of the workshop was designed to combine several academic skills which we believe every university graduate must be exposed to. These skills include:

- Accumulation and self-study of advanced material - The students are required to learn the material in the textbook on their own and to search for additional material on the internet and in other books.
- Preparation and delivery of a lecture in front of fellow students - Each student is assigned one of the topics of the workshop and must prepare an hour long lecture which will be delivered in class in front of the students and the course staff. The lecture is based on one of the chapters in the textbook and additional material which they have accumulated.
- Design and Programming of a large programming project - The projects are much larger than any other programs that the students handed-in through out their studies. This requires the ability to implement OOD and Programming skills.
- Collaboration among students - The project is handed-in in pairs so the students must learn to work with each other. The OUI is a distance learning institution therefore collaboration between students is often carried out through the network.
- Software installation - The students must cope with installation problems which may occur while installing and using software packages used in the workshop.
- Project presentation - The students are required to present their project in front of the course staff. The students must defend their project and demonstrate knowledge in all the topics covered in the workshop.

3. STUDENT COLLABORATION

As mentioned in the previous section the student lecture and final project are the main requirements in the workshop. The student lecture is performed individually by each student while the project can be performed by pairs. We consider collaborative programming an extremely important skill which should be developed in

undergraduate education. This is because software development in industry is often performed by teams of software engineers working collaboratively on projects [4]. It is known that programming projects performed in pairs are completed in a shorter time and produce better code than projects performed by individuals. In addition industry programmers working in pairs have reported higher job satisfaction than those who work alone [13].

The literature reports on benefits of collaborative programming in CS courses relative to individual programming: the students produced better programming assignments; they experienced a reduced rate of frustration; in some cases they received higher grades in the course; working in pairs reduced the amount of time the students had to spend on the assignments; and it also reduced workloads for the teaching staff [3, 8, 10, 14]. Some negative aspects regarding collaborative programming were also reported. These include difficulties in scheduling, incompatible partners [3], and unequal participation [12].

The collaborative programming discussed in several papers [3, 4, 8, 9, 10, 12, 14] takes place in undergraduate CS courses in traditional universities. Our collaboration experience takes place in an advanced CS course in a distance learning environment. The project is handed-in in pairs, therefore collaboration between students is often carried out through the network. Some of the pairs meet only once or twice, or not at all, during the work on the project due to the fact that they live in different geographical areas of the country. This style of collaboration through the internet is unique in distance learning environments.

The following section describes a research which we have been conducting at the OUI regarding the collaborative programming work which takes place in the workshop.

4. THE RESEARCH

The OUI is a distance learning institution, therefore collaborative programming raises some difficulties which do not exist in traditional universities. Our students are located all over the country, many of them are not acquainted with their peers and therefore finding a partner and working together is more difficult. Most of the CS courses at the OUI do not allow collaborative programming therefore the workshop is perhaps the only opportunity for such an experience. From our experience, backed by research mentioned above, we believe that there are many benefits both to students and staff which can be gained by working in pairs. In order to promote collaborative programming, the course staff strongly urges the students to find pairs for the final project. The students team up on their own without staff intervention. In some cases the students use the discussion group in the course web-site in order to find a suitable partner, in other cases they approach other students via e-mails which are published in the course web-site. Students who do not find a partner or prefer to work alone are permitted to do so, since working in pairs is not mandatory in the workshop.

During the academic year of 2004-5 we conducted a research in order to evaluate the benefits of collaborative programming in the workshop.
The research hypotheses we raised are presented below.

4.1 Research Hypotheses

H1. When given a choice, most of the students will prefer to work on the project in pairs.
H2. Weak students working in pairs will receive higher grades on the project compared to weak students working on their own.
H3. Students working in pairs will complete their projects earlier than those working alone.
H4. When given a choice, students tend to pair up with students on the same level, but in cases where stronger students pair up with weaker students the grades of the stronger students are not harmed significantly.

4.2 Research Population

The group of students we are analyzing is composed of 77 students which were enrolled in the course "Advanced Programming in Java Workshop" in the fall and spring semester of 2005. As mentioned above, all the students in the workshop completed the prerequisite Java course with a grade of 70 or above. Similar to all courses in the OUI, the students are located all over the country. The geographical distribution of the students in this course is that 81% of the students are located around the center of the country, while 19% of the students live in remote areas.

5. RESULTS AND DISCUSSION

This section describes the data we collected and analyzed in an attempt to shed some light on the research hypotheses listed above.

5.1 Pairing preference

The course staff highly recommends that the students will work in pairs. The benefits of team work are explained to the students. They are told that each student has different strengths (design, technical, programming etc.) and combining them will lead to better projects. Also, having a partner to consult with is very valuable. We assumed that the majority of the students will choose to work in pairs.
We checked the number of students who chose to work in pairs and we found that about 50% of the students (38 out of 77) chose to do so. 36% of the students (28 out of 77) chose to work alone, and the rest (14%) did not hand in the project and therefore did not complete the course.

We were somewhat surprised to find that 36% of the students preferred to work alone. This seems to be a high percentage but it may be explained by the distance education and the non typical student body at the OUI. Fifty percent of the students in the workshop were 30 years old or older. From conversations with the students they claimed that finding a partner was difficult because they were not familiar with other students in the course, they did not all live in the same geographical area and many of them worked and raised families.

We tried to further characterize the students in order to determine which of the students worked in pairs and which worked alone. We checked whether the students' level (indicated by their overall grade average) or their location of residence affected their decision to pair up or to work alone.

We divided the students into two groups according to their overall grade average: mediocre students with an average grade lower than 85, and good students with an average grade of 85 or above. The following table shows that out of the good students, a higher percentage of them (65%) preferred to work in pairs.

	Percentage of solo students	Percentage of paired Students
Mediocre students (Overall average < 85)	42%	58%
Good students (Overall average >= 85)	35%	65%

Table 1: Percentage of solo and paired students according to their overall average

When checking the residence area of the students in the workshop, we found that 19% of them live in remote areas of the country. We checked what their coupling preference was and found that most of them (66%) preferred to work in pairs compared to the 50% found in the course population. From this we can see that living in a remote area is perhaps an incentive to working in pairs. We also found that 26% of the couples were composed of students living in different geographical areas of the country. Working in pairs is possible in cases where students do not live in the same geographical area due to the distance education practiced at the OUI.

5.2 Success in the project

We wanted to check whether the students working in pairs will receive higher grades in their projects compared to students working on their own. In order to do so, we divided the students into two groups according to their overall grade average: mediocre students with an average grade lower than 85, and good students with an average grade of 85 or above. For each group we checked the project grade distribution achieved by the students who worked alone and compared it to the project grade distribution of the students who worked in pairs.

The following figure shows the grade distribution of the paired projects versus the solo projects for the good students:

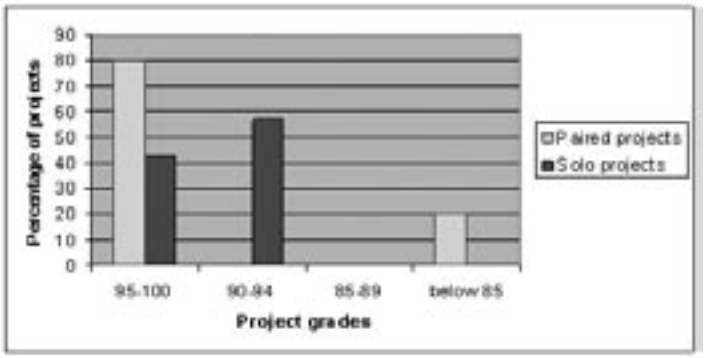

Figure 1: Project grade distribution for the good students - Solo versus Paired

From the graph one can see that almost all the good students received a project grade of 90 or above. Furthermore we can see that the good students who worked in pairs received grades higher than the good students who worked alone. 80% of them received a grade between 95 and 100, while only 43% of the good students who worked alone received this grade.

The next figure shows the grade distribution of the paired projects versus the solo projects for the mediocre students:

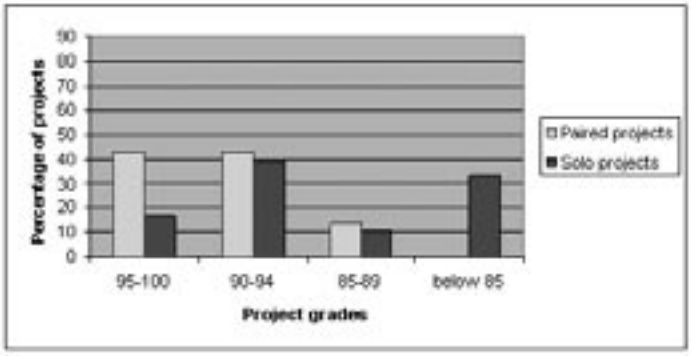

Figure 2: Project grade distribution for the mediocre students - Solo versus Paired

From this graph one can see that mediocre students who worked in pairs received higher grades than mediocre students who worked alone. 86% of them received a grade between 90 and 100, while only 56% of the mediocre students who worked alone received this grade.

The average grade of the projects both for the good students and mediocre students who worked in pairs was about 93. The average grade of projects for the good students who worked alone was 94, while the average grade of the projects for the mediocre students who worked alone was 88. These findings strengthen our hypothesis that mediocre students who worked in pairs received higher grades than mediocre students who worked alone.

Furthermore we noticed that in most cases, the grade each student received in the project was higher than his overall average. We were wondering if this grade gain was similar for students who worked in pairs compared to students who worked alone. We found that the average grade gain for paired students was 10.5 points while the average grade gain for solo students was 8.96 points. This finding also suggests that one can achieve a higher grade when working in pairs than when working alone.

5.3 Time to project completion

The project due date is set for two and a half months after the end of the semester. The due date is not very strict and students can ask for an extension which is granted if the students have made significant progress on their projects. We found that 74% of the couples handed in their projects on time, while only 61% of the students who worked alone handed in their projects on time. The couples completed their projects in an average time of 2.7 months while the solo students completed their projects in an average time of 3.7 months. This finding suggests that working in pairs contributes to completing the project on time. From conversations with the students we found that couples urge each other to hand in the project on time.

5.4 Partner selection criteria

We assumed that students will tend to pair up with students who are on the same level, that is good students will prefer good partners and weaker students will match up with students on their level. This assumption is based on the fact that good students do not want to take upon themselves the majority of the workload [4].

In order to determine the level of the students we decided to look at their overall grade point average. Another possibility which we considered was to look at their grades in the prerequisite course, but we decided that the overall average was a better indication of the student's level, because it is composed of all their courses up to that point. We found that the overall grade point average distribution of the students who worked in pairs ranges from 73 to 93. We also found that 74% of these students have an overall average grade between 80 and 89. The following graph shows the overall average grade distribution of students who worked in pairs:

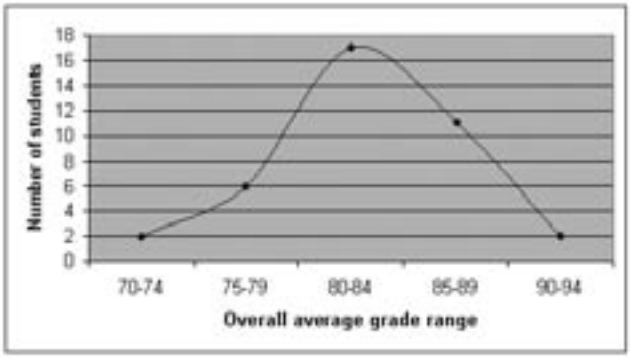

Figure 3: Overall average grade distribution of students who worked in pairs

We checked the overall averages of each couple in order to determine how close these averages were. We found that about half of the couples had overall average differences of up to 5 points and the other half had overall average differences of 6 to 11 points. Although, the overall average grades range from 73 to 93, no couple had an overall average difference larger than 11 points. This means that students did select partners who were more or less on a similar level as themselves. The following graph shows the differences between the overall average grades of each couple. The couples in the graph are sorted in ascending order according to the overall average grade differences - the first couple had no difference in their overall average grades while the last couple had 11 points difference in their overall average grades:

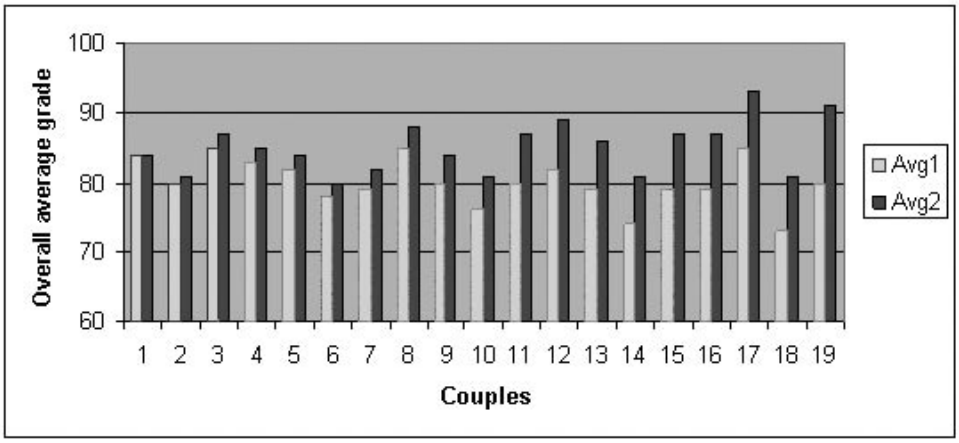

Figure 4: Overall average grades of couples

Furthermore we tried to analyze which of the students gained most by working in pairs. We defined the grade gain for each student as the number of points by which the project grade was higher than the overall average. We divided the pairs into two groups:

(1) Homogeneous group - Pairs with overall average grade differences of up to 5 points.

(2) Heterogeneous group - Pairs with overall average grade differences of 6 to 11 points.

Each pair is composed of a stronger and a weaker student. We checked the average grade gain of the stronger students and the weaker students in each group. Although it is obvious that stronger students will gain less than the weaker students, we were wondering whether this gain loss was significant.

We found that the average grade gain of the stronger students in the homogeneous group was 8.3, while the average grade gain of the weaker students in this group was 10.7. For the heterogeneous group we found, that the average grade gain of the stronger students was 7.7, while the average grade gain of the weaker students in this group was 15.6. Looking at these numbers, one can see that the students who gained the most from pairing up were the weaker students in the heterogeneous group (15.6). This grade gain was found to be significantly different ($p < 0.05$) from the grade gain of the weaker students in the homogeneous group (10.7). This result provides support for H2. On the other hand the stronger students in the heterogeneous group were not harmed very much by taking on much weaker partners. We can see that their average grade gain was 7.7 while the average grade gain of the stronger students in the homogeneous group was 8.3, which is not significantly different ($p > 0.05$). This finding strengthens our hypothesis H4 that in cases where stronger students pair up with much weaker students (6-11 points weaker), the grades of the stronger students are not harmed significantly.

6. CONCLUSIONS AND FUTURE WORK

We have reported on a collaborative programming research we have conducted in an advanced programming workshop in a distance learning environment. The final project of the workshop can be handed-in in pairs. Our study investigates several aspects of collaborative programming and we came up with the following findings:

- When given a choice, a larger percentage of the students work in pairs than alone. Further more, the percentage of good students who work in pairs is higher than the percentage of mediocre students who work in pairs. The students who live in remote areas of the country

exhibit a higher percentage of pairing than the rest of the students.

- The average grade in the project achieved by the couples is higher than the average grade re ceived by the students who worked alone. Furthermore, the average grade gain of the paired students is higher than the average grade gain of the solo students.
- The paired projects are completed in a shorter time than the solo projects.
- Students tend to pair up with students of the same level but when stronger students pair up with much weaker students, the grades of the stronger students are not harmed significantly. Although weaker students may have a hard time finding a partner, they will benefit most.

As we can see from this research, the benefits of collaborative work in the projects are apparent and therefore we plan to consider requiring the students to work in pairs. Today the students pair up according to their own preferences without staff intervention. As a result, some students remain without partners. We are aware that randomly assigning partners may lead to incompatible partners [10], but we will consider staff intervention in cases where students are unable to find partners. There are many factors which influence the success of collaborative work. Some of these factors include personality [15], skill level [4], availability etc.. We will try to take these factors into consideration when matching up students for the projects.

Today the division of work between the students in a couple is performed by the couple according to their preferences. We do not insure that the workload is divided equally among the students and therefore some students may pass the course without contributing significantly to their projects. We plan to devise means by which we can enforce equal work on the projects.

7. REFERENCES

[1] Benaya, T. and Zur, E. Distance Learning in Graduate School. *The 18th AAOU Annual conference*, November 2004.

[2] Benaya T. & Zur E. Traditional and Web-Based Distance Education in Computer Science Courses. I*CDE International Conference on Open Learning & Distance Education*, November 2005.

[3] Bevan, J., Werner, L. and McDowell C. Guidelines for the Use of Pair Programming in a Freshman Programming Class. *Proceedings of the 15th Conference on Software Engineering Education and Training*, p.100, February 25-27, 2002.

[4] Cliburn, D. C. Experiences with Pair Programming at a Small College. *Journal of Computing Sciences in Colleges*, 19(1), October 2003.

[5] Deitel, H. M., Deitel, P. J. and Santry, S. E. A*dvanced Java 2 Platform - How to Program.* Prentice Hall, 1st edition, 2002.

[6] Gal-Ezer, J. and Lupo, D. Integrating internet tools into traditional CS distance education: Students' attitudes. *Computers & Education* 38, 4, April 2003, 319-329.

[7] Guri-Rosenblit, S. *Distance and Campus Universities: Tensions and Interactions, A Comparative Study of Five Countries*, Pergamon, 1999.

[8] McDowell, C., Werner, L., Bullock, H. and Fernald, J. The effects of pair-programming on performance in an introductory programming course. *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, February 27-March 03, 2002, Cincinnati, Kentucky.

[9] Mendes, E., Al-Fakhri, L. B. and Luxton-Reilly, A. Investigating Pair-Programming in a 2nd-year Software Development and Design Computer Science Course, *ITiSCE 2005 - Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, June 2005.

[10] Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C. and Balik, S. Improving the CS1 experience with pair programming. P*roceedings of the 34th SIGCSE technical symposium on Computer science education*, February 19-23, 2003, Reno, Navada, USA

[11] The OUI Website: http://www-e.openu.ac.il/

[12] Thomas, L., Ratcliffe, M., and Robertson, A. Code Warriors and Code-a-Phobes: A Study in Attitude and Pair Programming. *Proceedings of the 34th SIGCSE technical symposium on Computer science education,* ACM SIGCSE Bulletin, 35(1), January 2003.

[13] Williams, L., Kessler, R., Cunningham, W. and Jeffries, R. Strengthening the Case for Pair Programming. IEEE Software, July/August, 2000.

[14] Williams, L., Yang, K., Wiebe, E., Ferzli, M., Miller, C. Pair Programming in an Introductory Computer Science Course. Initial Results and Recommendations, presented at the OOPSLA *Educator's Symposium*, Seattle, Washington, 2002.

[15] Williams, L. and Kessler, R. *Pair Programming Illustrated.* Boston, Massachusetts: Addison Wesley, 2003.