Initial findings on the impact of an alternative approach to Problem Based Learning in Computer Science

J. O' Kelly, S. Bergin, S. Dunne, P. Gaughran, J. Ghent, A. Mooney
Department of Computer Science, NUI Maynooth, Maynooth, Co. Kildare, Ireland

Abstract
A student on a programming module needs to know how to solve problems, design and test programs, learn the syntax of a programming language and possess good communication skills. We had previously identified that the reason why students experience problems with programming is due to their poor problem solving ability. To attempt to alleviate these problems we integrated an alternative PBL approach into the programming module[1]. In this paper we provide an analysis of the impact of our changes based upon qualitative and quantitative data gathered from interviewing and surveying all parties involved in the PBL process, notably lecturers in their capacity as module coordinator and problem creator, tutors in their capacity as facilitators of PBL workshops and students, including mature students, foreign students and repeat students. In addition, qualitative data gathered from the problem refinement process is presented. We believe that this research will be of particular interest to any Institution considering integrating PBL into an existing module.

1. Introduction
First year computer science courses tend to have high failure rates.[2][3] Prior to the commencement of the current academic year, the first year programming module consisted of three one-hour lectures and a three-hour lab each week, with optional periodic tutorials. A student's final mark was based upon continuous assessment, in the form of class and laboratory exams, and an end of year exam. Students taking this module come from various academic disciplines. Sometimes students did not know many other students taking the module and this often resulted in a feeling of isolation. Furthermore, it was observed that a major difficulty experienced by students was the transfer of information from the lectures to the labs, and specifically the problem appeared to be caused by an inability to break programming problems up into manageable steps.[4] In this paper we present our initial findings on the impact of our changes since the introduction of PBL. We outline our research methodology and present qualitative and quantitative data gathered from the students who participated in the module and the parties involved in co-ordinating the PBL sessions.

2. Research Methodology
Students participating in the introductory programming module come from three unique course disciplines: (1) Computer Science and Software Engineering (2) Computer Science through a Science degree and (3) Computer Science through an Arts degree. These students included: direct entry students, repeat students, mature students and foreign national students. In the academic year 2003/2004, 110 students participated in this module, 62% of the participants were males and 38% of the participants were female. To determine the impact of integrating PBL into the introductory programming module data was gathered through a number of different methodologies throughout the year. An end-of-year student questionnaire was returned by 80% of the class. We also interviewed 30% of the class consisting of 60% of mature students, 88% of repeat students, 88% of direct entry students and 50% of foreign students. Furthermore, during the course of the year students completed two other review forms while facilitators completed a review form each week for every workshop group. A weekly problem refinement meeting took place to gather information about the previous workshops and to prepare for the coming workshop. At the end of the year facilitators also completed questionnaires and the lecturer of the course was interviewed on the PBL process.

3. Initial findings of PBL co-ordinators
In our department, a lecturer is responsible for co-ordinating a module. First year lectures were typically held in a large auditorium to a large number of students from various academic disciplines. Sometimes, students only knew a very small number of other students taking the module and this resulted in a feeling of isolation, particularly for struggling students who feared that they were the only students who didn't understand. Moreover, students tended not to ask questions at lectures. Observations by both the lecturer and the module

tutors indicated that the major difficulty experienced by students was the transfer of information from the lectures to the labs.[4] PBL was integrated into the module in an effort to reduce the high attrition rate and to alleviate the frustration and isolation felt by students. The lecturer in their capacity as module co-ordinator, believes that the PBL workshops give students an opportunity to work as part of a team, to get to know the team members well and thus to reduce the feeling of isolation and frustration. PBL has encouraged them to ask questions in the workshops and this has resulted in them asking questions in lectures. Finally, PBL has helped students to learn how to break problems up into manageable components and explore alternative solutions.

Before the introduction of PBL lab problems were used to reinforce the constructs covered in lectures. Initially, problems were very simple, however, as the course progressed they became more complex as new concepts were introduced. Students always worked alone to solve a problem and if a student didn't grasp a particular concept then they often struggled in the lab. The introduction of PBL has resulted in a conscious change in the types of problems used. Abstract problems that require students to work in a group environment were introduced. As part of the group process students have to explain their ideas to other students. The process of verbalising their thoughts helps to either clarify their understanding or to realise a problem with their thought process. In addition, they have learned to apply the process of breaking a problem up into steps to other lab programming problems. Finally, the types of problems now used allowed students to be more creative, to take risks, but also to recognise the constraints of their knowledge of programming and of the programming language they are using.

Prior to the introduction of PBL, tutors assisted at the weekly labs and gave periodic tutorials. The tutors observed that the length of the labs (three hours) resulted in fatigue and diminished productivity levels for both the students and the tutors as the lab progressed. Additionally, tutors observed that students would often try to solve a programming problem by calling over one tutor after another, soliciting a new piece of information each time. As students often did not know or were not very familiar with other students in the lab, they rarely asked other students for help. The integration of PBL into the programming module has meant that a three hours lab is now split into a one and a half hour workshop followed by a one and a half hour lab. The facilitators of the workshops noticed that although initially students looked solely to them for help in solving a problem, they have now learned to help each other. Finally the restructuring of a three-hour lab has alleviated the problems of fatigue.

The introduction of a problem-refinement process allows the lecturer and facilitators an opportunity to look at a problem from a student's perspective and to identify potential issues with the problem prior to the workshop. The problem refinement meetings were in effect PBL workshops; the group worked through the problem to determine if solving it would satisfy the required learning outcomes. From this process the group have learned that the best problems are ones that are easily extendible [1]. Additionally, these meetings enabled the group to identify any issues and implement the subsequent required changes.

4. Initial findings from PBL participants

We present an analysis of data gathered from the PBL participants in four main areas: the weekly workshops, the impact of group work, student performance and the perceived role of the facilitators. In addition, we outline some other interesting issues from the gathered data.

4.1 Workshops

The purpose of introducing the PBL workshops was to increase student motivation and develop their problem solving skills. Of the 30% of students interviewed, 81% found the workshops to be very good or beneficial. Furthermore, 85% of students felt they had a better understanding of problem solving and of the programming language, and that their confidence had increased as a result. 100% of the repeat students interviewed agreed that the workshops were a valuable addition to the course, as did all of the mature students. The repeat students' felt that they had gained a more in-depth knowledge of programming due to the workshops and felt that the workshops had helped to alleviate the feeling of isolation that they had previously experienced.

4.2 Groups

The students were divided into 17 groups in each semester. It was anticipated that the individual students would see the advantage of working together in a group. All of the groups found the group environment more conducive to learning. When interviewed, 78% stated the workshops were helpful, opened their minds to other aspects and ideas of problem solving, promoted teamwork and the open discussion of viewpoints. 85% of groups interviewed, found the process of solving the problem as a group more educational than finding a solution. 71% of groups professed an interest in taking another module that used a PBL approach. The majority of groups found the inclusion of props helpful in the process of problem solving.

4.3 Student Performance

We rated a student's effectiveness in the workshop by their ability to support their beliefs, have effective communication skills, participate in the group, be open to new ideas and show constructive critical thinking. This suggests that if a student demonstrates the abilities described above then the student would have the capability to solve and program traditional computer science problems. We found that there was a correlation between the students that performed well during the workshops and the top 10% of the class that performed well in the lab. However, it was found that the students that performed poorly in the lab did not necessarily perform poorly in the workshop. This result suggests that if a student achieves good lab results, he/she should achieve good workshop results but the converse hypothesis is not necessarily true.

It was found that students that performed well in the role of the chair also performed well in the workshop. The top 10% of the class that performed well in the role of the chair had a mean percent of 85.45% in this role while their mean workshop result was 85.99%. However, there exists no correlation between how well a student performs in the lab and how well a student performs in the role of the chair.

4.4 Role of facilitators

Qualitative data has been gathered on the role of the facilitators in the PBL process, both from the students and the facilitators' perspective. The facilitators felt that they had a good understanding of their role as facilitators and that they helped to develop a student's reasoning process. Data gathered from the students validated the facilitators' beliefs and furthermore showed that the facilitators demonstrated good judgement in providing necessary information and knowing when to deflect questions back to the group. The students also thought that the facilitator had an impact on helping them learn, while the facilitators considered themselves to be more passive and felt that the students learned through their own initiative and their interaction with the group.
The restructuring of groups in the second semester resulted in a number of students having a new facilitator with some experiencing a facilitator working with two concurrent workshop groups. The majority of students did not encounter any problem with a change in the facilitator or with the groups they were in.

4.5 Other Issues

As the number of male students outweighed the number of female students taking the module we were concerned that gender might have been an issue in the PBL workshops. As discussed in [1] in forming each group we tried to maintain gender balance. Our findings indicate that gender was not an issue with 96% of the students stating this.
As mentioned in Section 4.4 the workshop groups were restructured in the second semester, as two dysfunctional groups were identified. While a small number of students were apprehensive at first about changing groups, the vast majority of students found changing the groups to be advantageous to them as they experienced other people's ideas and problem-solving approaches.
There was a range of views professed by groups relating to formative assessment, particularly regarding what students should be assessed on and who should be doing the assessment. 92% of groups found group peer-to-peer assessment very beneficial, helping to build confidence within the team and encourage competition among groups. 68% of groups queried, thought the PBL groups should be formatively assessed, although no agreement was reached as to how they should be assessed or the proportion of marks that should be allocated.

5. Discussion

As this is our first year to use PBL we are not in a position to make any substantive claims. However, our initial findings are very positive and it is reasonable to state that the lecturers, facilitators and the majority of students feel that the introduction of PBL is worthwhile and beneficial. An intangible effect that we have observed is the support mechanism that has resulted from the PBL workshops. Through these workshops, students had the opportunity to get to know each other better, and to recognise how other students approach and represent a problem. We believe that this has been of significant benefit to the students. Another interesting and pleasing finding of this study was that the facilitators did not value their contribution to a student's learning as highly as the students did.

Continuous assessment marks indicate a significant improvement in student performance in comparison to last year's marks, as detailed in Table 1.

Continuous Assessment	2004	2003
Percentage of students passing (Pass mark is 40%)	84.00	59.13
Average class mark	58.67	45.59
Average mark top 10% of class	89.33	80.00
Average mark bottom 10% of class	25.00	15.00

Table 1: Comparison of Continuous Assessment Marks 2003 - 2004

After the end of year exam, we would like to carry out further analysis to determine if the introduction of PBL has had a positive effect on a students overall performance. Furthermore, we have observed a correlation between average and above average student's performance in workshops and their continuous assessment. We intend to investigate this correlation further. If we can confirm that such correlation does exist we may be able to develop a diagnostic tool to predict how we expect students to perform in a programming exam given their performance in the weekly workshops. This may enable us to make interventions at an earlier stage to prevent students from failing.

6. Conclusion

In this paper we presented the initial findings on the integration of PBL into an existing introductory programming module. While we cannot make any substantive claims based on this data, our initial findings are very positive. We intend to carry out further analysis on the data gathered and to use a similar approach in the next academic year.

References

[1] O'Kelly J et al. **An Overview of the Integration of Problem Based Learning into an existing Computer Science Programming Module**. Pleasure By Learning, (PBL 2004), Cancun, Mexico

[2] Morgan, M., Flanagan, R., Kellaghan T., **A Study of Non-Completion in Undergraduate University Courses** (2001)

[3] Robins, A., Rountree, J., Rountree, N., **Learning and Teaching Programming: A Review and Discussion.** Computer Science Education, (2003) Vol. 13, No. 2, pp. 137-172

[4] McCracken M et al **A multi-national, multi-institutional study of assessment of programming skills of first-year CS students.** Report by the ITiCSE 2001 Working Group on Assessment of Programming Skills of First-year CS students.