

Applying a hybrid problem-based learning method to the teaching of computer programming

Yakun Wu

Computing Center
Liaoning University
66 Mid-Chongshan Road
Huanggu District
Shenyang Liaoning 110036
People's Republic of China

Wuyakun2001@sina.com

Mentor

Bing Bing Zhou
School of Information
Technologies
Faculty of Science
The University of Sydney

Abstract

This paper first presents the main features of problem-based learning (PBL). Then, it describes a hybrid PBL model which explains how PBL could be applied to a course in computer programming. A discussion of the requirements for successful introduction of these changes is given.

Introduction

Many important changes have taken place in the educational models used at universities outside the People's Republic of China (PRC) over the last few decades. Innovative teaching methods, such as PBL provide new and interesting models of teaching aimed at improving student learning. One of the most important aspects of PBL is the role of students in the learning process as PBL places the student at the centre, this is different from the traditional approaches still common in the PRC in which the teacher is generally placed at the centre. In PBL students take command of their own learning and are required to improve working cooperatively and their communication skills as well as their abilities of analysis and synthesis. Furthermore, because PBL promotes self-directed learning this tends to improve the likelihood of the student to engage in lifelong learning. Since McMaster University Medical School first pioneered the method of PBL [1] in 1966, this method has been widely adopted in the tertiary sector and a great variety of PBL applications have been reported in tertiary education. In the following sections of the paper, the main features of PBL will be described first followed by a possible way that PBL could be applied in a course of computer programming. Finally, some conclusions and discussions are presented.

The problem-based learning method

What is PBL?

PBL is a curriculum design and a teaching/learning strategy which simultaneously develops higher order thinking, disciplinary knowledge, and practical skills by placing students in the active role of practitioners (or problem solvers) confronted with a situation (i.e., an ill-structured problem) which reflects the real world (Illinois Mathematics and Science Academy 1994). The main idea of PBL is to use common, discipline-specific problems as a motivational starting point to encourage learning. In its pure form (or full PBL), students are first presented a problem with no previous preparation. The problem is usually designed to be ill-structured, complex, open-ended and reflecting the real world. The students discuss and analyse the problem, and list what they know and, we would hope, why they need to know the items on the list. They determine and locate the resources to be used. Then, they learn the knowledge independently, and try to apply them to solve the problem in groups. Finally, they summarise their work, present the problem and justify their solutions. Their performance is evaluated by themselves, members of their groups and teachers. In this process, students become responsible for their own learning.

Models of PBL vary considerably. Ellis, Carswell and Bernet divide the PBL methods into three categories. The simplest form is the problem-based approach, in which the material is presented in normal lectures, but problems are used to motivate students and demonstrate the theory. The second form is a hybrid model or guided PBL. In this case, problems are solved in groups, but also lectures are used to present the fundamental concepts and some of the more difficult topics. The third one is a full PBL, where the problems guide and drive the entire learning experience; in this form there are no lectures from the 'expert' and groups or individuals work independently of one another.

Since there are different kinds of PBL models, it is useful to indicate the common characteristics that are shared by most models. The following are some of the main characteristics which are typical for problem-based courses:

- use of 'real-life' problems to engage the students in the learning process;
- include content that crosses the boundaries between disciplines;
- emphasise students taking responsibility of their own learning;
- focus on the process of knowledge acquisition rather than the final product itself;
- change the role of instructor to facilitator;
- involve self-assessment and peer-assessment of student learning; and
- focus on communication and interpersonal skills.

Why adopt the PBL method for teaching computer science?

The majority of students entering university in China come directly from secondary or high-school education, where the education system is examination-oriented. The students are placed in a highly pressured environment and the competition to achieve a place in university can be extreme. The selection process is assessment driven which requires students to take learning as a task in which they memorise, store and reproduce information. This promotes rote-learning, commonly called 'surface learning', that doesn't match well with the requirements of education (or the workplace) in the 21st century. The rapid explosion of knowledge requires that students be self-directed and adaptable – two of the more important, necessary skills for lifelong learning. Increased technology and social complexity needs students to be able to integrate interdisciplinary knowledge and to be involved in teamwork and have well-developed interpersonal skills.

Besides this, 'surface learning' is not a suitable learning strategy in computer science, as the ability to solve problems is a vital element of this discipline (and most others). Computer science matches the characteristics of PBL in the following ways: (Ellis, Carswell and Bernet)

- computing is, for the most part, problem driven;
- lifelong learning is a necessity due to the rapidity and continually changing nature of the industry; for instance, the program you use for a particular task today probably won't be the one you use for that same task in a decades time;
- practitioners must constantly update their skills and competencies in order to keep abreast of new technology;
- project group is the predominant mode of operation within the industry; and
- computing crosses discipline boundaries.

The Information Technology (IT) industry is thus constantly changing. To be effective in this changing environment requires the professional to be able to constantly reassess and modify their strategies. This reflection is built into the PBL process.

Applying hybrid PBL model to the course of computer programming

The course description

Computer Programming is a service course which is taught in the second semester of the first year. The total number of hours allocated to the course is 64, including 32 hours lectures and 32 hours laboratories. The students have one two-hour lecture and one two-hour laboratory session each week. Tutorials are available during the laboratory times. Students taking the programming course come from totally different disciplines: Philosophy, History, Arts, Chinese, Law, and Science. At present it is taught with a traditional teacher-centred approach. The objective of the course is to cultivate in students the basic knowledge and basic skills about programming, so that they can apply the knowledge to solve some simple real-world problems.

We find that our course in computer programming is very difficult for the first year students and they tend not to do very well. Every year about thirty percent of the students fail the examination. One of the major reasons is that students lack the ability to analyse and solve problems. When a problem is presented to them, they usually don't know where to begin. They can't find a suitable algorithm to solve the problem. According to my teaching experience, I think that this situation is mostly due to the teaching approach. Usually we begin our lectures with the introduction of the syntax of a particular programming construct. Then, it is demonstrated in isolation and later incorporated into a larger program that solves a particular problem. Students are able to understand the construct in isolation and recognise it in the sample program but are unable to transfer this knowledge to their own programming. It seems to me from my experience in our classes that PBL offers an effect way to overcome this problem. So I have planed to try the PBL teaching method in the first year programming course when I return to teach at my university. The overall objective is to improve the students' ability of analysing and solving problems, to alter the pattern of learning from one of 'surface learning' to one in which 'deep learning' is enhanced. I also hope to encourage students to be responsible for their own learning.

Course design

Ellis, Carswell and Bernet indicate that first-year students making the transition from a teacher-centred school environment to a more self-directed university environment may need the comfort of a well-defined problem with considerable scaffolding. So I decide to use a hybrid PBL model in my teaching instead of the full PBL model. The major differences between them are: the duration of the problems, the continued inclusion at least one lecture every week and the method of assessment which still includes traditional examinations.

Because of timetabling constraints I am unable to change the time allotted to the course, but I can change the approach and methods I use in lecture time. In the new course design, students will still have a two-hour, traditional practical exercise session weekly, but the lecture time will be scheduled in a new way. Some of the lecture

time will be used in opening and processing problems, and others will be given over to traditional lecturing.

In the new course design, the whole process of teaching will be divided into two phases. In the first phase, which will last about eight weeks, I will adopt a problem-based approach. In the following phase, I will shift to a guided PBL approach.

In the problem-based approach, material will still be presented in lectures, but I will use a problem to construct a reality for the student that gives meaning to the concepts being learned. For example, suppose I introduce the topic of 'selection'. In the traditional teacher-centred approach, the selection construct will be presented first. Then, the teacher will explain how it works in detail followed by an example. In the problem-based approach, a problem is presented to motivate the concept first. In the case of selection, the simplest problem can be 'enter two pieces of data and compare them'. The nature of the problem can be discussed, with an emphasis on helping students understand why and how the selection process aids the process of comparing. The details of the construct are left for the students to investigate for themselves. In this setting, students will often complete a more traditional individual assignment as part of the assessment of their understanding.

Guided PBL will begin about the middle of the semester and continue till the end of semester. Firstly, students are given some real-world problems, then, they will be grouped into teams to solve problems. Since the students already have some basic programming knowledge and techniques, I will design some larger and much more complicated problems for them. Some possible examples of the problems could be the management of 2008 Beijing Olympics, traffic management on Dongling Road, or a video shop database. Groups are asked to choose one of these problems. At this stage, students are required to develop and implement plans for researching, designing and coding solutions to the chosen problem. Pace is maintained by a requirement to provide reports on their progress and activities regularly. Finally, I will ask students to reflect on their learning and present their work to the rest of the class and for assessment. In this case, the problem drives the students' needs for knowledge construction. Lectures will be used to present fundamental concepts in the problem development, and a range of resources will be available to provide assistance in detailed knowledge acquisition during the learning process.

Assessment

In the hybrid model, students can be evaluated by both summative assessment and formative assessment. Since it is the first time using a PBL approach, and to avoid losing control of the teaching process, the model still includes some small quizzes scattered in the whole semester, and students' attendance to the laboratory tutorials will be recorded. Besides this, a final examination is still kept as a method to evaluate the students. In addition to the summative assessment, student performance in the projects will be evaluated in a formative way. With the project going, groups will give presentations and reports about the project, including project's integrated prototype,

demonstrate their projects and give walkthroughs of their code. According to the problem reports, their presentations to the class, their demonstrations, a formative assessment mark is given. The main assessment criteria for this part are as follows:

- how well the group is able to follow simple instructions?
- how much understanding of the problem is shown?
- how complete the prototype is?
- how clearly the group is able to communicate?
- how well the system performs during the demonstration?
- how easily the system can be used? and
- how clear the implementation is, how easily it is to understand the code and the purpose of each part?

The distribution of the marks is summarised in Table 1.

Table 1. Distribution of marks

Task	Maximum marks
Quizzes	40
Project	60
Attendance	0
Final Examination	100

From the table we can see that no marks are given to attendance. In fact, students' attendance in the laboratory tutorial is regarded as a factor which will affect the continuous assessment of students. The continuous assessment mark that is finally awarded to a student depends on the raw marks from the quizzes and the project, which is then multiplied by the attendance factor. It is initially defaulted to 1.0. The attendance factor will be decreased (see Table 2) if the number of absence rises, which leads to a lower overall mark. The final mark a student receives is the average number of the continuous assessment mark and the mark for the final examination.

Table 2. Values of attendance factor

Number of Absence	Attendance Factor
≤ 2	1.0
3	0.8
4	0.7
5	0.6
≥ 6	0.5

Discussion and conclusions

In this paper, a model for the course of computer programming is presented, which uses a hybrid form, after Ellis, Carswell and Bernet, of the PBL approach. The hybrid PBL model provides a good transition for students to a university environment. The model provides a framework to assist the students in problem abstraction, problem analysis, and problem solving. It helps develop their critical thinking skills, their verbal and written communication skills and their ability to work in groups.

The reality of an academic's experience of higher education in China today is that students do not seem to be as interested in learning as they once were. By employing active learning strategies, such as PBL, students will not only learn the knowledge that we as educators perceive to be important, but develop their ability to be independent learners and develop the skills required for lifelong learning. This makes them more competent in the rapidly changing knowledge society, and makes them not only better students in later courses, but also better professionals in the future. Yet the adopting of the new teaching method may mean challenges for both students and faculties.

Firstly, the students are accustomed to be taught in the traditional way, they generally expect to be told what to do and to have all the answers given to them. Changes will make them uncomfortable and they should be induced to know why their learning style must change. It will take time for them to adjust themselves to the new way of teaching.

Secondly, from the teachers' point of view, PBL requires teachers to change their role: from transmitter/instructor to mentor/coach. This involves some difficulties: they have to develop coaching skills. There is also a change in the structure of class time, and the teachers have to redesign the course before it starts. Teachers must spend most of their time on planning the PBL content and sequence of assignments, providing immediate feedback on students' work, and discussing and evaluating students. As teachers we might not get it right the first time but we will need to alter our teaching practice – as someone said during our classes – in a form of continual evolution.

Acknowledgement

First, I am very grateful for the China Scholarship Council for providing the opportunity to attend the *Teaching Sciences in English Program* at The University of Sydney, I have had a colourful experience and have learnt a lot. Then, I would like to give my thanks to Associate Professor Mike King and the teachers from Science Faculty who gave us occasional educational seminars outlining a wonderful variety of contemporary teaching methods. I also appreciate the help from the English teachers from the Centre for English Teaching, especially Asha and Cecilia. Besides, I will give my sincere thanks to my mentor Bing Bing Zhou, who gave me a lot of help both in work and life. Last, but not least, I wish to thank all my dear classmates for the happy times we have experienced together in Sydney.

References

- Beaumont, C., Sackville, A. and Cheng, C. *Identifying Good Practice in the use of PBL to teach computing*, [Online] Available: www.ics.heacademy.ac.uk/italics/Vol3-1/PBLGoodPracticePaperV9.htm.
- Ellis, A., Carswell, L. and Bernat, A. (1998) Resources, Tools, and Techniques for Problem Based Learning in Computing. In *Working Group reports of the 3rd annual SIGCSE/SIGCUE ITiCSE conference on Integrating technology into computer science education*. ACM Press, 41–56.
- Garcia-Famoso, M. (2005) Problem-based learning: a case study in computer science. *m-ICET2005* [Online] Available: <http://www.formatex.org/micte2005>.
- Greening, T., Kay, J., Kingston, J.H. and Crawford, K. (2000) Results of a PBL Trial in First-Year Computer Science. ACM International Conference Proceeding Series; Vol. 2 archive *Proceedings of the 2nd Australasian conference on Computer Science Education*, 1–20.
- Hadjerrouit, S. (1999) A Constructivist Approach to Object-Oriented Design and Programming. In *Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education*, 171–174.
- Hämäläinen, W. (2004) *Problem-based learning of theoretical computer Science*. 34th ASEE/IEEE Frontiers in Education Conference October 20–23, 2004, Savannah, GA.
- Martí, E., Gil, D. and Julià, C. (2006) A PBL Experience in the Teaching of Computer Graphics. *Computer Graphics forum*, **25**(1), 95–103.
- McConnell, J.J. (1996) *Active learning and its use in Computer*. Annual Joint Conference Integrating Technology into Computer Science Education Proceedings of the 1st conference on Integrating technology into computer science education, Barcelona, Spain, 52–54.
- Nuutila, E., Kinnunen, P., Törmä, S. and Malmi, L. *PBL Cases and Their Role in Learning Programming*, [Online] Available: http://www.spop.dk/chapters-abstractv1/en_pk_st_lm.pdf.
- Nuutila, E., Törmä, S. and Malmi, L. (2005) PBL and Computer Programming – The Seven Steps Method with Adaptations. *Computer Science Education*, **15**(2), 123–142.
- O'Kelly, J. (2005) Designing a hybrid problem-based learning (PBL) course: A case study of first year computer science in Nui, Maynoot. In T. Barrett, I. Mac Labhrainn, and H. Fallon (Eds) *Handbook of Enquiry & Problem Based Learning*. Galway: CELT.
- Sanwar, A. (2005) Effective teaching pedagogies for undergraduate computer science. *Mathematics and Computer Education*, Fall.