SIMULATOR BUILDING AS A PROBLEM-BASED LEARNING APPROACH FOR TEACHING STUDENTS IN A COMPUTER ARCHITECTURE COURSE

By

LI-MINN ANG * KAH PHOOI SENG **

** Associate Professor, University of Nottingham, Malaysia.*
*** Associate Professor, University of Nottingham, Malaysia.*

ABSTRACT

This paper presents a Problem-Based Learning (PBL) approach to support and promote deeper student learning in a computer architecture course. A PBL approach using a simulator building activity was used to teach part of the course. The remainder of the course was taught using a traditional lecture-tutorial approach. Qualitative data was collected using a questionnaire survey to evaluate the effectiveness of the approach. The student responses showed that the PBL approach was successful in developing deeper learning experiences and general skills such as time management and teamwork while at the same time contributing to a more enjoyable learning experience. As part of the investigation, the authors also looked at student fears and concerns as they were asked to make a transition from a more structured to a less structured learning environment.

Keywords: Problem-based learning, Constructivist learning, Computer architecture, Simulator.

INTRODUCTION

A well-known distinction on student learning is between the surface and deep learning approaches. Deep learning approaches promote critical analysis of new ideas, linking facts to known concepts, long term retention of knowledge and information and problem solving in unfamiliar contexts. Surface learning relies on memorization, tacit acceptance of information and superficial retention of material for examinations (Ramsden, 1992). Problem-based learning (PBL) approaches are often associated with deep learning. PBL was first applied for teaching medical students at McMaster University in 1969 (Barrows and Tamblyn, 1980). This learning approach has since been applied for teaching students in many disciplines such as business (Stinson and Milter, 1996), information technology (Cheong, 2008), law (Moust, 1998) and engineering (Linge and Parsons, 2006). This paper focuses on an application of PBL in engineering education. In PBL, learning is the result from the process of working towards the understanding and resolution of a problem in a real-world context. The conventional lecture-tutorial approach is teacher-centered where students learn by acquiring and absorbing information from teachers. The PBL approach is student-centered where learning is through the act of problem-solving. In PBL, students are presented with a situation that leads to a problem for them to solve. This teaching approach requires students to analyze the given problem, gather relevant information, propose and evaluate candidate solutions and then select and justify their final solution.

At the University of Nottingham in Malaysia, the authors teach an elective course on computer architecture to fourth year electrical and computer engineering students. In 2007, they decided to trial a PBL approach to teach part of the course. The motivation for adopting a PBL approach was that because they observed that the traditional lecture-tutorial approach was producing only superficial learning in some of their students which in some cases was not retained after the final examinations. The new course was divided into two parts. The first part taking up two-thirds of the course was taught using a traditional lecture-tutorial approach. The remaining one-third of the course was taught using a PBL approach. For the PBL learning process to be effective and to maintain student motivation, the problem or task to be solved has to be chosen to be grounded in the curriculum, authentic and mirror real-world activities. On the one hand, students should have the opportunity to bring in their own ideas, experiences and approaches to accomplish the task. On the other hand, because PBL activities would often require more effort from students, it is important that students have the perception that the task is a worthwhile problem for them to spend their time on. For both of these reasons, the authors decided to adopt a simulator building problem for the PBL activity.

Software simulators are useful tools in education to teach complex concepts. A few examples are the BRAINTRAIN (Panchaphongsaphak, et al. 2007), Microworld (Kato, 2006), SimTeacher (Fischler, 2007)and PSpice (Hart, 1993) simulators to teach medical, business, education and engineering students respectively. Simulators allow students to better visualize the situation and interactively explore the modeled domain. For example, the BRAINTRAIN simulator allows medical students to touch a physical model and see which parts of the brain respond to the touch. For most disciplines, students learn by using the simulator only. The authors call this as "simulator-using" activity. For computer engineering students, the development and construction of software is an integral part of the degree program. There is an opportunity to create a deeper learning experience if students are involved not only in using the software simulator but also contributes to building parts of the simulator itself. This then becomes a "simulator - building" activity. Student learning will be achieved from both the "simulator-using" and "simulator-building" activities. The simulator building activity will also allow students an opportunity to exercise their software engineering skills that were acquired in the earlier years of the degree program.

At the end of the course, the authors collected qualitative data using a questionnaire survey to evaluate the effectiveness of the PBL approach. The student responses showed that student learning processes can be deepened and accelerated by creating effective combinations of lectures-tutorials and PBL activities. Students also reported that the activity helped them to learn general skills such as time-management and team work and also contributed to a more enjoyable learning experience. As part of the investigation, the authors also looked at student fears and concerns as they were asked to make a transition from a more structured to a less structured learning environment. One of their findings on student concerns is that PBL activities could have an adverse effect on learning if students perceive that the effort required from them does not commensurate with the credits received for the course.

1. Problem-Based Learning Process with Simulators

PBL activities require a carefully planned process to guide the students through complex tasks such as brainstorming, identifying useful knowledge, formulating appropriate research questions and working out strategies for finding solutions. PBL is a constructivist learning approach (Tam, 2000) which proposes that learning occurs when students make connections of new information with their previous knowledge and experiences. Students discover new knowledge as they study and analyze the problem, propose possible solutions and reach a final result. The role of the teacher is to be a facilitator to enable and foster the constructive activities to take place. Figure 1 shows a framework for the PBL learning process which is adapted from (Tan, 2003).

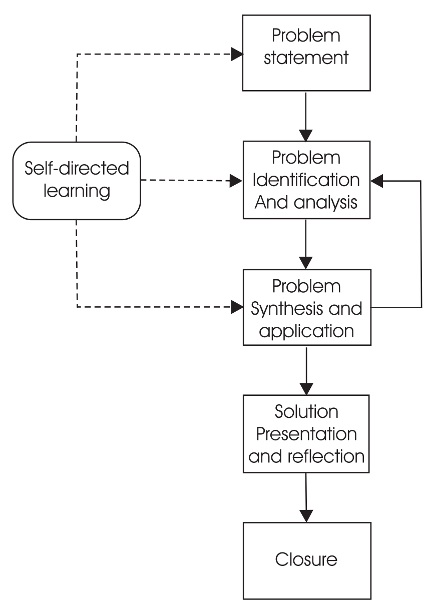

Figure 1. Framework for PBL learning process.

In the first stage, the facilitator presents the students with the problem statement to be solved. The facilitator may give an introductory mini-lecture to acquaint the students with the material to be learnt. The students are divided into groups and asked to reflect on the problem and gather the necessary background material. The students then progress to the second stage of problem identification and analysis where the groups brainstorm, identify, analyze and formulate an appropriate research question to be solved. They also identify the learning issues to be performed through self-directed learning. The facilitator guides the students and ensures that they are on the right track. In the third stage of problem synthesis and application, the gathered information is synthesized and applied to solve the problem. The facilitator checks to ensure that the students have covered the problem area sufficiently. The stages of problem analysis and synthesis are often an iterative process where students re-evaluate their solution after discovery of new information and ideas. The fourth stage is when groups present and reflect on their solutions. The facilitator asks questions to probe the depth of the students understanding. In the final stage, the facilitator closes the learning process by summarizing important principles and concepts and clears up any doubts and concerns which may arise from the students.

Simulators have been used as teaching tools in engineering education. A recent study was conducted at the Universidad del Valle, Colombia where lecturers used a wireless communications simulator to teach mobile communications to undergraduate students (Vejarano and Guerrero, 2008). In their methodology, senior undergraduate students develop simulators of telecommunication technologies under the guidance of an advisor during the last year of their studies. These simulators are then used in the undergraduate courses to improve students understanding of these technologies. Another study reports on a learning approach using simulators which had been used in an Instruction Level Parallelism Processors course (Moreno, et al., 2007). In this methodology, the students are divided into groups and each group is assigned a case study which is analyzed with the simulator, after which the students discuss the results with the lecturer. In both of these studies by Vejarano and Moreno, the role of the simulator for teaching undergraduate students is only for analysis and to obtain experimental results. The authors call this as "simulator-using" activity. Because the authors are teaching computer engineering students, they liked to involve them in a deeper learning experience by also contributing to building parts of the software simulator itself. This then becomes a "simulator-building" activity. Student learning will be achieved from both the "simulator using" and "simulator-building" processes.

2. Research Methodology

The objective of the computer architecture course is to help students gain a good understanding of the operation of high performance computing technologies. Students who enroll in this course have a background in software programming and digital systems. The teaching for the course was divided into two parts. The first part taking up two-thirds of the course was taught using a traditional lecture-tutorial approach. The remaining one-third of the course was taught using a PBL approach using simulator building as the learning activity. Table 1 shows the three modules in the course and the teaching approach used. Each module was taught in a period of four weeks. The second module on Pipelined Microprocessors was taught using a PBL approach. In this module, there are three main topics to be covered which are the pipelined microprocessors, data hazards and control hazards.

To introduce the PBL sessions, students were first informed on the learning objectives and the reasons for using the PBL approach. An overview of Bloom's taxonomy for learning (Bloom, 1956) was also included and students were encouraged to pursue deep learning. Then the students were given a mini-lecture on the basic concepts behind pipelining in computer systems. To put a mental picture into the students' minds and to trigger their thought

Module	Title	Approach
1	Machine Instruction set Architecture	Lecture-tutorial
2	Pipelined Microprocessors	PBL
3	Memory Systems	Lecture-tutorial

Table 1. Course modules and teaching approach

processes on pipelining, an everyday example using a fast-food drive through operation was presented. Students were asked a question on why they thought fast-food restaurants usually used three counters (one counter for the customer to order food, one counter to pay for the food and a third counter to collect the food) to serve their drive-through customers instead of just using a single counter to perform the entire operation. By reflecting on this analogy, the students gain an intuitive understanding of pipelining techniques and are ready to begin the simulator building activity.

The students were then divided into groups and given the problem statement which was to build a simulator of a pipelined microprocessor in software. The pipelined microprocessor simulator to be built was based on the MIPS microprocessor architecture (Patterson and Hennessy, 2007). To guide the construction of the simulator, the students were also given three datasets A, B and C to test their simulator with. These data sets were carefully constructed in terms of increasing complexity to achieve the desired learning outcomes. Table 2 shows the three data sets and the desired learning outcomes. The students were not informed of the purposes of the data sets and what specific conditions the different data sets were testing for. They were only informed that the three data sets were ordered in terms of increasing complexity.

During the first session, a basic simulator with minimal functionality was introduced to the students and the students were asked to run the simulator with Dataset A. Dataset A works with the basic simulator. After confirming that Dataset A works, the students were then asked to run the simulator with Dataset B which does not work with the basic simulator. The students have to brainstorm and analyze why Dataset B does not work when Dataset A

Dataset	Complexity	Desired learning outcome
A	Low	Understand and analyze operation of basic pipelined microprocessor - hazard-free architecture.
B	Moderate	Analyze, evaluate and resolve data hazards in pipelined microprocessor
C	High	Analyze, evaluate and resolve load and control hazards in pipelined microprocessors

Table 2. Datasets and desired learning outcomes.

does, gather the relevant information from books and technical papers, and make programming modifications so that their improved simulator will work with Dataset B as well as with Dataset A. If they achieve this, the students would have learnt to analyze, evaluate and resolve data hazards in pipelined microprocessors. This learning process is repeated for Dataset C where students teach themselves and their group members to analyze, evaluate and resolve load and control hazards in pipelined microprocessors. On completion, the students add enhancements to the simulator and submit the work as their coursework assessment. It is interesting to note that although the groups were all given the same problem statement and datasets for testing, their approaches and final solutions were very diverse in range. Three groups used a complex hardware structure for the processor with a corresponding simpler software program. The remaining two groups took the opposite approach and produced a simpler hardware processor structure at the expense of a more sophisticated software program.

One consideration for the simulator building activity is whether to build a text-based or a graphical-based simulator. It was decided to go with a text-based simulator because the authors had concerns that building a graphical simulator may cause students to be bogged down with unnecessary programming details for the graphical components. The simulator allows the students to view the contents of the registers and pipeline information in the microprocessor at each clock cycle. This information allows them to identify exactly when the microprocessor begins to go wrong so that they can propose and make the necessary modifications to the simulator. This learning activity engages the students in both the "simulator-using" and the "simulator-building" process. Students learn visually by using the simulator. At the same time, they also learn kinesthetically by building it. The activity also contributes to a more enjoyable learning experience. On their own initiative, a few groups added their own extra features to the simulator and made improvements to the user interface.

3. Student Responses and Findings

An important point in the investigation is how to gather useful data. First, the authors had a small sample of ten students. Second, they recognized that students are busy people and they are only interested in getting authentic responses. For these reasons, they decided to use a qualitative questionnaire survey approach using only the four questions shown in Table 3. The questions are open-ended and the students are free to respond in any way that they wish. The first two questions are concerned with the students' learning preferences and the last two questions are concerned with the effectiveness of the learning approach from the students' own viewpoints.

Table 4 shows a summary of the student likings towards the PBL activity. It is worthwhile to note the words the students have selected to frame their responses in. Four groups used phrases associated with visual learning such as "greater insight", "see the bigger picture", "clear picture of what we were learning" and "see how the processor is running". The fifth group framed their response with a phrase associated with hands-on learning when they said the activity "made it easy to grasp concepts behind machine language programming". It is interesting to note that the simulator building activity helped the fifth group understand better the material taught in the first course module on the Machine Instruction Set Architecture. This is one of the objectives of deep learning where students are able to link material they have learnt from one module with material learnt from a previous module. The student responses gives evidence that deep learning has taken place either through the visual process (visual learning) or through the hands-on process (kinesthetic learning).

Questions
What did you LIKE about the PBL Learning Sessions?
What did you NOT LIKE about the PBL Learning Sessions?
What did you LEARN from the PBL Learning Sessions?
What did you NOT LEARN from the PBL Learning Sessions?

Table 3. Survey questions.

Group	Likes
1	Insterting.. Greater insight into what was being studied in the lecture
2	Helpful.. Helped me to see the bigger picture of machine language and other computer operations.
3	Well guided throught the sessions .. Done ina step by step manner which made it easy to grasp concepts behind machine language programming
4	Gave us a clear picture of what we were learning and also recognize how to apply theory in practice
5	Instead of words in notes, I can 'see' how the processor is running and getting results.

Table 4. Responses on student likes.

Table 5 shows a summary of the student dislikes. There are two main themes. The first theme is to do with the time allocated for the activity. Phrases such as "needed more time", "one hour wasn't adequate time", "time consuming" and "session is too short in time" were received. In future, the authors will schedule the PBL sessions such that students will have a longer two-hour session instead of two one-hour sessions weekly. The second theme is hinted at by Group 4 which said that the "time required to complete the task doesn't really match up to 15% worth of assignment". One can almost hear the thought process behind those words, "I like it but it is too expensive! The amount of work required is not worth doing for 15%". As lecturers for the course, the authors agree with them. A fair course work assessment for the time and efforts required by the simulator building activity would be 30%. If the work was valued at 30% for the course assessment, the authors' view is that most of the student dislikes would go away. However, because the same course is currently taught on two university campuses (Malaysia and UK), this will require subject lecturers from both campuses to commonly agree to a change in the course assessment structure.

Table 6 and 7 shows a summary of what the students said they learnt or did not learn respectively. The students report on learning general skills such as time-management and

Group	Dislikes
1	Needed more time in the labs with the lecturer
2	I did not bother about the sessions until due date of assignment was near
3	One hour wasn't adequate time. Just us you felt you were getting the hang of it we had to stop and that was not welcome
4	Task was a bit too time consuming. Which doesn't really match up to 15% worth of assignment
5	The session is too short in time. Perhaps it can extend to 2 hour session

Table 5. Responses on student dislikes

Group	Learnt
1	How the pipelining simulation works and how we can use this to our advantage
2	Everything I need to know to get my project to take off
3	It helped us plan out our assignment and take it step by step rather than leaving us lost in the middle
4	Gave us a chance to familiarize with matlap programming as well
5	No response

Table 6. Student responses on what they learnt.

Group	Did not learn
1	Trying to visualize the processor as (software) code was a very difficulttask to perform transferring from paper to code or from a diagram tocode especially when not knowing matlab (programming) very well
2	It is through the project which helped me and my partner to work better.This we call 'teamwork' was not learned in the sessions but in the process of handling up the final project
3	We felt that we could have had a few more project learning sessions to become even more familiar with the relevant topics.
4	Nothing in particular
5	No response

Table 7. Student responses on what they did not learn

teamwork. One group (Group 1) commented that it was a "difficult task to perform transferring from paper to code especially when not knowing matlab programming very well". Unfortunately, these students have forgotten about their software programming skills which were acquired in the earlier years of the engineering program. However, another group (Group 4) commented that the PBL activity "gave us a chance to familiarize with matlab programming".

The authors also looked at how well the students performed for the final examinations. After marking the examination papers, the authors observed that for questions where the students have choices, they have some preference for the questions related to the PBL learning sessions. They also gained higher marks for those questions in relation to other questions. This gives some additional evidence for the simulator building PBL activity to support and promote deep learning. Another observation was that students could answer the more difficult parts of the questions when it was related to the PBL activity. When asked to describe the inner workings of the pipelined microprocessor, a few students could even support their answer by writing relevant software code in their examination paper. This behavior was not seen in previous year's students' examination scripts. Additional benefits of the sessions which were conveyed verbally to the authors by some students was that they enjoyed the additional time for interaction with the lecturers, felt a sense of achievement after completing the real-world simulator task and reduced the monotony of attending traditional lectures daily.

When initially considering using a PBL approach, the authors had some considerations about students' fears and concerns because they were asking them to make a transition from a more structured to a less structured learning environment. This was one reason why the authors selected the pipelining module to be taught using the PBL approach because it was sandwiched between the other two traditional lecture-based modules. Throughout the PBL sessions, the authors monitored and observed the students for signs of anxiety. A clear indicator would be seen if a student decided not to come to class for the PBL sessions. However, the authors observed the opposite when they had almost full attendance for all the PBL sessions. On the whole, they observed that the students' attendance for the PBL sessions were higher than for the traditional lecture based sessions. Furthermore, on entering the classes for the PBL sessions, the authors noted that some groups had already begun loading their simulators and were waiting for the instructors to arrive so that they could have a discussion. The student responses showed that the main student concerns were not with having to do a new activity or with the less structured teaching approach employed. A few students even responded with comments of "very enjoyable". Their main concerns were to do with the time and effort that they had to spend on the task. In particular, they felt that the amount of time that they had to spend on the task was not commensurate with the credits received for the task. One finding from this study is that the tasks and corresponding credits received for a PBL activity have to be carefully considered so that students feel that they are getting a fair assessment.

One group commented that they would have liked a "few more learning sessions to become even more familiar with the relevant topics". From the viewpoint of the instructors, this may not be practical because of the time constraints and also because of the additional demands on the students' time. The authors found that using one-third of the teaching time available for PBL activities and the remainder of two-thirds for traditional lecture-based teaching gave a good balance. Finally, it has been noted that planning the PBL sessions required some initial effort on the part of the instructors. The basic simulator which was the starting point for the activity took the instructors and a senior undergraduate student six months to develop.

Conclusions

They are two main conclusions from this study. First, the authors have shown that student learning processes can be deepened and accelerated by creating effective combinations of lectures-tutorials and PBL activities. The simulator building sessions helped students to see the bigger picture and made it easy for them to grasp complex concepts. The sessions also gave students an opportunity to practice software programming skills acquired in earlier years of their course, promoted group activity which helped them to learn general skills like time management and teamwork, increased interaction between students and instructors and contributed to a more enjoyable learning experience. Secondly, they experienced that PBL activities could have an adverse effect on learning if students perceive that the effort required from them does not commensurate with the credits received for the assessment.

References

[1]. **Barrows, H. S. and Tamblyn, R. H. (1980),** *Problem-based Learning: An Approach to Medical Education,* New York, Springer Publishing.

[2]. **Bloom B. S. (1956),** *Taxonomy of Educational Objectives, the Classification of Educational Goals Handbook I: Cognitive Domain*, New York, McKay.

[3]. **Cheong, F.** (2008), "Using a problem-based learning approach to teach an intelligent systems course", *Journal of Information Technology Education*, 7, 47-60.

[4]. **Fischler, R. (2007),** "SimTeacher.com: an online simulation tool for teacher education", *TechTrends*, 51, 44-47.

[5]. **Hart, D. W. (1993),** "Circuit simulation as an aid in teaching the principles of power electronics", *IEEETransactions on Education,* 36, 10-16.

[6]. **Kato, Y. (2006),** "An XML-based microworld simulator for business modeling education", *Proceedings Fourth International Conference on Creating, Connecting and Collaborating through Computing*, 232-239.

[7]. **Linge, N. and Parsons, D. (2006),** "Problem-based learning as an effective tool for teaching computer network design", *IEEE Transactions on Education*, 49, 5-10.

[8]. **Moreno, L., Gonzalez, C., Castilla, I., Gonzalez, E. and Sigut, J. (2007),** "Use of constructivism and collaborative teaching in an ILP processors course", *IEEE Transactions on Education,* 50, 101-111.

[9]. **Moust, J. (1998),** "The problem based education approach at the Maastricht law school", *The Law Teacher*, 32, 5-36.

[10]. **Panchaphongsaphak, B., Burgkart, R. and Riener, R. (2007),** "Three-dimensional touch interface for medical education", *IEEE Transactions on Information Technology in Biomedicine*, 11, 251-263.

[11]. **Patterson, D. A. and Hennessy J. L. (2007),** *Computer Organization and Design: the Hardware/Software Interface*, Morgan Kaufmann.

[12]. **Ramsden, P. (1992),** *Learning to Teach in Higher Education,* London, Routledge.

[13]. **Stinson, J. E. and R. G. Milter. (1996),** "Problem-based learning in business education: curriculum design and implementation issues", *New Directions for Teaching and Learning*, 33-42.

[14]. **Tam, M. (2000),** "Constructivism, instructional design and technology: implications for transforming distance learning", *Educational Technology and Society*, 3, 50-60.

[15]. **Tan, O. S. (2003),** *Problem-based Learning Innovation: Using Problems for Power Learning in the 21st Century,* Thomson Learning, Singapore.

[16]. **Vejarano, G. and Guerrero, F. G.** (2008), "A constructivist simulation-based methodology for teaching mobile communications", *IEEE Transactions on Education*.

ABOUT THE AUTHORS

** Li-minn Ang is currently an Associate Professor at the University of Nottingham , Malaysia. His research interests are in the areas of image and video compression, computer and reconfigurable architectures for image processing, intelligent processing techniques and learning approaches for Engineering Education.*

*** Kah Phooi Seng is currently an Associate Professor at the University of Nottingham, Malaysia. Her research interests are in the areas of intelligent visual processing, biometrics and multi-biometrics systems, artificial intelligence, signal processing and the application of technology tools for education.*