

# Puzzle-Based Learning for Engineering and Computer Science

**Nickolas Falkner**

School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia  
*jnick@cs.adelaide.edu.au*

**Raja Sooriamurthi**

Information Systems Program, Carnegie Mellon University, Pittsburgh, PA 15213, USA  
*raja@cmu.edu*

**Zbigniew Michalewicz**

School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia  
*zbyszek@cs.adelaide.edu.au*

***Abstract:** A number of universities are introducing courses, modules, or freshman seminars that relate to the puzzle-based learning paradigm. The purpose of such units is to attract, motivate and retain engineering/computer science students, and to increase their mathematical awareness and their problem-solving skills. This is essential in developing students as general problem-solvers, rather than training students to limit their thinking to the problems that are immediately pertinent to the specific area that they are studying. The business world has already recognised that problem-solving skills are valuable and some major companies actively recruit good puzzle-solvers. We introduce and define our puzzle-based learning approach, provide an example of the syllabus and course material and provide evidence of two current implementations, including early student feedback.*

## Introduction

A recent article describes a puzzle-based freshman seminar introduced at the University of California, Santa Barbara, to motivate and retain computer engineering students (Parhami, 2009). The author argues that attracting students to computer science and engineering programs represents only one aspect of a broader problem of the shortage of a skilled information technology workforce, and that recruitment efforts must be augmented with additional strategies for retaining and motivating students – strategies that are missing in curricula recommendations of the IEEE Computer Society and the ACM.

The problem may be even broader. Today’s marketplace needs graduates capable of solving real problems of innovation and a changing environment – we need more skilled graduates. What is missing in most engineering/computer science curricula is coursework focused on the development of problem-solving skills. Most engineering students never learn how to think about solving problems in *general* – throughout their education, they are constrained to concentrate on textbook questions at the back of each chapter, solved using material discussed earlier in the chapter. This constrained form of “problem solving,” is not sufficient preparation for addressing real-world problems. On entering the real world, students find that problems do not come with instructions or guidebooks. One of our favourite examples to illustrate this point is a puzzle on breaking a chocolate bar:

*A rectangular chocolate bar consists of  $m \times n$  small rectangles and you wish to break it into its constituent parts. At each step, you can only pick up one piece and break it along any of its vertical or horizontal lines. How should you break the chocolate bar using the minimum number of steps (breaks)?*

If you do not know the answer, which textbook would you search to discover the solution? The same applies to solving many real world problems: which textbook should you search to find a solution, if that is the solution strategy that you've learned?

Students often have difficulties in independent thinking or problem-solving skills regardless of the nature of a problem. At the same time, educators are interested in teaching “thinking skills” rather than “teaching information and content.” The latter approach has dominated in the past. As Alex Fisher wrote in his book (Fisher, 2001): “... *though many teachers would claim to teach their students ‘how to think’, most would say that they do this indirectly or implicitly in the course of teaching the content which belongs to their special subject. Increasingly, educators have come to doubt the effectiveness of teaching ‘thinking skills’ in this way, because most students simply do not pick up the thinking skills in question.*”

Further, many analysts lament the decreasing mathematical skills of students. A recent *Mathematics Working Party Final Report*, issued by the University of Adelaide (June 2008) includes statements such as “*There is an urgent need to raise the profile and importance of mathematics among young people...*” and “*The declining participation in mathematics and related subjects is not limited to Australia...*”.

Our universities, The University of Adelaide and Carnegie-Mellon University, have introduced a new course titled Puzzle-based learning to address all of the issues raised here.

## The puzzle-based learning approach

The puzzle-based learning approach aims at encouraging engineering/computer science students to *think* about how to frame and solve *unstructured* problems - those that are not encountered at the end of some textbook chapter. Our goal is to motivate students, and also to increase their mathematical awareness and problem solving skills by discussing a variety of *puzzles*. The course is based on the best traditions introduced by Gyorgy Polya and Martin Gardner over the last 60 years (Polya, 1945; Gardner, 1961).

Many teachers have used puzzles for teaching purposes and the puzzle-based learning approach has a much longer tradition than just 60 years (Danesi, 2002). The first mathematical puzzles were found in Sumerian texts circa 2,500 BC. However, some of the best evidence of the puzzle-based learning approach can be found in the works of Alcuin, an English scholar born around AD 732 whose main work was *Problems to Sharpen the Young* – a text which included over 50 puzzles. Some twelve hundred years later, one of his puzzles is still used in artificial intelligence textbooks to educate computer science students.<sup>1</sup>

In this course we concentrate on *educational puzzles* that support problem-solving skills and creative thinking. These educational puzzles satisfy most of the following criteria:

1. *Independence*: The puzzles are not specifically tied to a single problem-solving domain.
2. *Generality*: Educational puzzles should explain some universal mathematical problem-solving principles. This is of key importance. Most people agree that problem solving can only be learned by solving problems, however, this activity must be supported by strategies provided by an instructor. These general strategies would allow for solving new yet unknown problems in the future.

---

<sup>1</sup> The puzzle is the “river crossing problem”: *A man has to take a wolf, a goat, and some cabbage across a river. His rowboat has enough room for the man plus either the wolf or the goat or the cabbage. If he takes the cabbage with him, the wolf will eat the goat. If he takes the wolf, the goat will eat the cabbage. Only when the man is present are the goat and the cabbage safe from their enemies. All the same, the man carries wolf, goat, and cabbage across the river. How has he done it?*

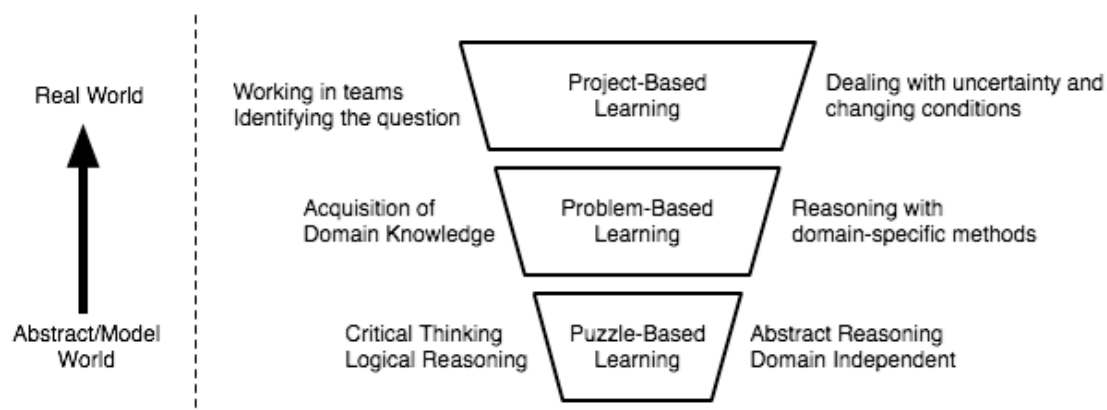
3. *Simplicity*: Educational puzzles should be easy to state and easy to remember. This is also very important, as easy-to-remember puzzles increase the chance that the solution method, including the universal mathematical problem-solving principles, is also remembered.
4. *Eureka factor*: Educational puzzles should initially frustrate the problem-solver, but with the promise of resolution. A puzzle should be interesting because its result is counter-intuitive: problem-solvers often use intuition to start their quest for the solution and this approach can lead them astray. Eventually a *Eureka!* moment is reached (Martin Gardner's *Aha!*), when the correct path to solving the puzzle is recognized. The *Eureka* moment is accompanied by a sense of relief, the frustration that was felt during the process dissipates and the problem-solver may feel a sense of reward at their cleverness for solving the puzzle. The *Eureka* factor also implies that educational puzzles should have elementary solutions that are not obvious.
5. *Entertainment factor*: Educational puzzles should be entertaining and engaging. Entertainment is often a side-effect of simplicity, frustration, the *Eureka* factor, and an interesting setting (e.g. a casino environment, fighting against dragons, or dropping eggs from a tower).

Educational puzzles can play a major role in attracting students to computer science and engineering programs and can be used in talks to high school students and during open-day events. Puzzles can also be a factor in retaining and motivating students. Above all they are responsible for developing their critical thinking and problem-solving skills as well as raising the profile and importance of mathematics. Further, there is a strong connection between the ability to solve puzzles and the ability to solve industry/business problems. Many real world problems can be perceived as large-scale puzzles. William Poundstone investigated the purpose of famous Microsoft interview puzzles and he wrote (Poundstone, 2000):

*At Microsoft, and now at many other companies, it is believed that there are parallels between the reasoning used to solve puzzles and the thought processes involved in solving the real problems of innovation and a changing marketplace. [...] When technology is changing beneath your feet daily, there is not much point in hiring for a specific, soon-to-be-obsolete set of skills. You have to try to hire for general problem-solving capacity, however difficult that may be. [...] Both the solver of a puzzle and a technical innovator must be able to identify essential elements in a situation that is initially ill-defined. It is rarely clear what type of reasoning is required or what the precise limits of the problem are."*

## Puzzle-based learning vs. Problem-based learning vs. Project-based learning

The ultimate goal of puzzle-based learning is to lay a foundation for students to be effective problem solvers in the real world. At the highest level, problem solving in the real world calls into play three categories of skills: dealing with the vagaries of uncertain and changing conditions; harnessing domain specific knowledge and methods; critical thinking and applying general problem solving strategies. These three skill categories are captured in the three forms of learning depicted below:



**Figure 1 A continuum of learning and skills needed for problem solving in the real world**

In this continuum, each layer of skills builds upon the layers below it. The focus of puzzle-based learning is on domain independent, transferable skills. In addition, we aim to foster introspection and reflection on the personal problem solving process. What was I thinking? What is the solution? Why did I not see it?

Both problem-based learning and project-based learning are well established methodologies (Blumenfeld et al. 1991, Bransford et al. 1986). By our description above, problem-based learning requires significant domain knowledge. This is the form of learning one typically sees emphasized in a domain specific undergraduate course such as electromagnetism, data-structures, or circuit-theory etc. Project-based learning on the other hand deals with complex situations where usually there is no one clear unique or correct way of proceeding, for example: How can we increase the adherence of cystic-fibrosis patients to follow their treatment protocol? It can be very hard to determine the best solution. The pedagogical objectives of project-based learning include dealing with ambiguity and complexity, integration of a variety of approaches, user-testing of the value of proposed solutions, and working with a team of people with diverse backgrounds and skills. In both problem-based and project-based learning the problem drives the learning: students need to assess what they already know, what they need to know to address the problem, and how to bridge the knowledge/skill gap.

Puzzle-based learning focuses on domain independent critical thinking and abstract reasoning. This leads to the question: What is the difference between a puzzle and a problem? One way of characterizing the difference is the extent to which domain specific knowledge is needed to solve it. The general flavor of puzzles is that their solution should only require domain neutral general reasoning skills – a biologist, a musician, and an artist should all be able to solve the same puzzle. The different styles of reasoning required for problem-based learning and puzzle-based learning could be compared to the difference between an in the field investigator and an armchair detective – one only requires reason.

An example that compares and contrasts problem-based learning and puzzle-based learning is the well known egg-drop experiment. The traditional problem-based learning version of this experiment involves finding a way to drop an egg from a maximal height without breaking. A puzzle-based learning version of this experiment also involves the dropping of an egg from a building but the question under investigation, though related, is quite different. In the problem-based learning approach a series of physical experiments are conducted to determine a way to maximize the height from which an egg can be dropped without breaking. There are two broad approaches: dampen the effect of the impact (leading to padding-based solutions) or lessen the impact (leading to delivery mechanisms, parachutes etc). The team-based learning outcomes of such an experiment are to determine different ways by which an impact can be dampened or lessened.

A puzzle-based learning approach to a similar problem does not involve a physical experiment but rather a thought experiment. One approach would be to ask a question along the lines of: using multiple eggs, what would be an effective strategy by means of which I could *determine* the highest floor from which I could drop an egg without breaking. There are interesting variations to this question: In this thought experiment there are three entities (i) the number of eggs (ii) the number of drops and (iii) the number of floors. One puzzle-based learning question could be: given a fixed number of eggs, and a number of allowed drops what is the maximum height of a building whose breaking floor we can determine? This could be denoted as  $F_{e,d}$ . An alternate question could be: given a fixed number of floors (say 100), and a number of eggs (say 3) what is the maximum number of drops needed to determine the breaking floor ( $D_{e,f}$ ). Yet another version could be to ask how many eggs would be needed to determine the breaking floor given a fixed number of floors and allowed number of drops ( $E_{f,d}$ ).

Note that all three puzzle-based learning versions of the problem require only basic math skills and analytical reasoning. One of the goals of puzzle-based learning is to foster the skill of analyzing and understanding problems clearly. Part of this requires the clarification of any assumptions needed to solve the problem. For example, for the egg-drop thought experiment some reasonable assumptions

such as “all eggs behave the same way” and “an egg that survives a drop takes no damage and may be dropped again”. To constrain the problem we would consider assumptions such as “an egg surviving a drop from floor  $x$  would have survived a drop from any floor less than  $x$ .”

Given a fixed number of eggs ( $e$ ) and a specified number of drops ( $d$ ), we want to determine the maximal height of a building whose breaking floor we can determine ( $F_{e,d}$ ). Applying the heuristic of “try to solve a similar but simpler problem,” let us consider the situation where we have only one egg. In this case, we are required to search sequentially. If we are allowed 10 drops ( $d=10$ ), then we can determine the breaking floor of a 10 floor building by starting at floor 1 and working our way up. Now suppose we had two eggs, what strategy could we follow? Students who have had some prior programming experience often give *binary search* as a possible strategy, although this is not the best solution. Students are led through the reasoning process in a lecture environment and are encouraged to contribute and refine their suggestions, with controlled prompting. By considering examples and reasoning about what happens if the first egg breaks, or if it doesn't break, students are guided through the general version of this puzzle culminating in the derivation of the general solution.

Puzzle-based learning shares many of the pedagogical goals of the emerging paradigm of Computational Thinking (Wing 2006). Puzzle-based learning resonates with the Computational Thinking emphasis on abstraction and analytical thinking. With reference to Figure 1, Computational Thinking straddles the whole problem skill spectrum but places more emphasis on Problem-based and Project-based learning. With its emphasis on domain independent, rigorous and transferable reasoning we believe that puzzle-based learning lays a basis for CT in the curriculum.

## The puzzle-based learning courses: types, structure, and content

There are a few different versions of the puzzle-based learning course being taught currently. The course can be offered as a full-semester (three units) elective course (typically 3 contact hours per week, split into lectures and tutorials), a full-semester (three units) freshman seminar (3 contact hours per week), one unit freshman seminar, and one unit core module as part of some other course.

One of the important points about puzzle-based learning courses is that the course is not about presenting and discussing a variety of puzzles but rather about presenting, discussing, and understanding problem-solving principles and some mathematical principles in the context of puzzles that serve as entertaining illustration of the presented concepts. Also, the process of understanding problem-solving principles leads students through a variety of topics, exposing them to many important concepts at early stages of their college education.

Despite a variety of possible offerings of puzzle-based, the structure of the course is very much the same. The topics listed below correspond to weeks of a 12 –week semester – regardless whether each topic is allocated one hour or three hours:

1. **Introduction:** What it is all about?
2. **The problem:** What are you after?
3. **Intuition:** How good is it?
4. **Modeling:** Let's think about the problem a bit more
5. **Some mathematical principles:** Do you see it?
6. **Constraints:** How old are my children?
7. **Optimization:** What is the best arrangement?
8. **Probability:** Coins, dice, boxes, and bears
9. **Statistically speaking:** What does it mean?
10. **Let's simulate:** Can we generate the answer?
11. **Pattern Recognition:** What is next?
12. **Strategy:** Shall we play?

Each topic is illustrated by a variety of puzzles presented in an interactive manner. The course introduces a few simple problem-solving rules that we refer to in every class. Every week students are presented with homework assignments: one or more puzzles on the topic covered in the class. The following week, at the beginning of a class, the solutions are presented and discussed. Homework contributes 30% towards the final grade, and the final exam contributes the remaining 70% of the final grade. Students have access to all lectures slides, audio lecture recordings, and additional material, including course software. In the following, we give some examples of a sample lecture, some homework and exam questions.

## **A sample lecture: “The problem: What are you after?”**

The lecture introduces the most important problem-solving rule:

**Rule #1:** Be sure you understand the problem, and all the basic terms and expressions used to define it.

Indeed, without understanding the problem, all our efforts in finding a solution are usually a waste of time. The use of *under-specification* can be used as a tool to encourage students to determine what they know, what they don't know and need to find out and what they are unable to find out. This places the emphasis for knowledge discovery on the student and also forces them to accept that, on occasion, one must provide the "best guess". The very first puzzle we use to illustrate this simple observation is one of the favourite puzzles of Martin Gardner:

**Puzzle #1.** *A farmer has:*

*20 pigs,  
40 cows, and  
60 horses.*

*How many horses does he have, if he calls the cows horses?*

It takes students a short while to understand the problem and that “calling” has little to do with “having”. The farmer still has 60 horses.

This example can be followed by the classic:

**Puzzle #2.** *You drive a car at a constant speed of 40 km/h from **A** to **B**, and on arrival at **B** you return immediately to **A**, but at a higher speed of 60 km/h. What was your average speed for the whole trip?*

Again, most students would jump immediately into the obvious answer of 50 km/h without much understanding of what the average speed is (or rather, how average speed is defined). For most of them, it is a surprise to discover the correct answer of 48 km/h – next time (in any other course they will take in their programs) they will think twice before they answer a question on some average. It is this clear and thoughtful analysis that we seek to foster, to hone and guide intuition.

Clearly, there are strong connections between the process of understanding the problem and critical thinking. A lecture may include a slide with a statement containing loose terminology, straw-men arguments and logical fallacies, and students are asked to discuss it. We seek to encourage thought, with a focus on critical thinking towards a solution.

One of the weekly assignments given to the students at the end of this lecture was:

**Homework #1.** *With a 7-minute hourglass and an 11-minute hourglass, find the simplest way to time the boiling of an egg for 15 minutes.*

Note that we are not after *any* solution, but the *simplest* solution. One week later, after all homework has been handed in, the lecturer has interesting material for discussion, as some students found the solution A:

- Start with the 7- and 11-minute hourglasses, when the egg is dropped into the boiling water.
- After 7 minutes, invert the 7-minute hourglass.

- After 4 additional minutes (i.e. when sand stops in 11-minute hourglass), invert the 7-minute hourglass again.
- When the sand stops in the 7-minute hourglass, 15 minutes will have elapsed.

whereas other students found solution B:

- Start the 7- and 11-minute hourglasses.
- After 7 minutes, drop the egg into the boiling water.
- After 4 additional minutes (i.e. when sand stops in 11-minute hourglass), invert the 11-minute hourglass.
- When the sand stops in the 11-minute hourglass again, 15 minutes will have elapsed from the time the egg was dropped into the water.

But which of these solutions is simpler? Solution A takes 15 minutes to complete and requires two inversions. Solution B requires 22 minutes, but only one inversion. Most students believed that solution A was simpler, as it required reduced time, however, they were less certain about that when they were told that the hourglasses involved were quite heavy – weighing 100 kg each. Such a puzzle provides excellent material not only for a discussion on understanding the problem: what does *simplest solution* mean? This also introduces the concept of multi-objective optimization, a concept students are usually exposed to during the 3<sup>rd</sup> year of studies.

## Some other examples

There are many puzzles that we can use to illustrate fundamental concepts of probability, statistics, pattern recognition, games, constraint-processing, optimization. A few examples are presented here:

**Puzzle #3.** *A farmer sells 100kg of mushrooms for \$1 per kg. The mushrooms contain 99% moisture. A buyer makes an offer to buy these mushrooms a week later for the same price. However, a week later the mushrooms would have dried out to 98% of moisture content. How much will the farmer lose if he accepts the offer?*

This is a very good example for resisting immediate intuitive answers – it might not be that obvious that the farmer will lose \$50.

**Puzzle #4.** *A bag contains a single ball, which is known to be either white or black (with equal probability). A white ball is put in, the bag is shaken, and a ball is then randomly removed. The removed ball happens to be white. What is now the probability that the bag currently contains white ball?*

Puzzle #4 introduces basic concepts in probability.

**Puzzle #5.** *There are  $n$  dots on a plane (flat surface). There are two players, **A** and **B**, who move alternatively; **A** moves first. The rules of the game are the same for both players: at each move they can connect two points, but they cannot connect points which were already directly connected to each other or connect a point with itself. In other words, they build a graph (with predefined  $n$  vertices) by connecting some of the dots. The winner is the one who makes the graph connected (a graph is connected if there is a path between any two nodes of the graph, however, not every two nodes have to be connected directly). What is the winning strategy for player **A**, if such exists?*

Puzzle #7 is a good puzzle to introduce graphs and to investigate the concept of strategies – it is not trivial to discover the winning strategies of the first or second player.

A few hundred educational puzzles were collected and organized into meaningful subsets. All teaching materials and the new textbook (*Puzzle-Based Learning: Introduction to Critical Thinking, Mathematics, and Problem Solving*) are now in active use – the text follows the structure of the course

given earlier. Chapter 13 of the text includes a collection of problems without a solution – these can be used for homework, assignments, and exams.

There are many ways to evaluate the progress of students in the puzzle-based learning course. These vary from evaluations based on participation through evaluations based on homework to final exams. Many students might be a bit nervous on sitting a final examination loaded with puzzles; however, there are many ways to organize the exam in a meaningful way. For example, last semester the final exam questions included:

**Exam Question #1.** *Five minutes after midnight of 13 April 2004 there was a heavy rain in Melbourne. 72 hours later, what is the probability that it would be sunny over there? Justify your answer.*

This checked their skills of understanding the problem before providing any answer. (For those who are curious, the probability is 0%.)

**Exam Question #2.** *The hour and minute indicators of my watch cover each other every 65 minutes. Is my watch running too quickly or too slowly? Justify your answer.*

This question was a test of their modelling skills and students were rewarded for identifying the implicit question, which is “when should the hands of a watch cover each other”, and for modelling the problem, even if they reached an incorrect conclusion.

As a glimpse of the range of approaches to a problem consider #2. In class we examined different styles of reasoning: Quantitative vs. Qualitative vs. Intuitive vs. Wild Guess. Both Quantitative and Qualitative are rigorous - one uses numbers and algebra and the other uses language and logic. One group of students went through the precise calculations as to when an overlap should occur on a correct watch, while another group qualitatively reasoned that if a clock were correctly running, then 60 minutes past noon the minute hand would be over 12 and the hour hand over 1. Five minutes later the minute hand would be over the 1 and the hour hand would have moved a little forward. From this point, students could still discuss different interpretations of the result, encouraging thought.

For more information on the nature of puzzles and the approaches used in puzzle-based learning, readers are directed to the website associated with the text, [www.PuzzleBasedLearning.edu.au](http://www.PuzzleBasedLearning.edu.au). We now discuss two of the implementations of this course, at the primary development site, The University of Adelaide and at Carnegie-Mellon University, Pittsburgh.

## **First experiences: The University of Adelaide**

The initial implementation of puzzle-based learning was a 1-unit course set as a component of a 3-unit first-year course, *Introduction to Electronic Engineering*. After support from the Dean of the Faculty of Engineering, Computer and Mathematical Sciences, from 2009, this 1-unit offering was placed into introductory courses across the engineering programs offered within the University. A 3-unit first-year course for students planning to major in computer science was launched simultaneously in 2009, and made available to all non-engineering students in the University. We refer to the 1-unit offering as PBL-E (PBL for Engineers) and the 3-unit offering as PBL Main. The courses cover the same material, at different levels of depth. The intake for the two courses is quite different, as PBL-E students have a higher Tertiary Entrance Rank on average and also have two courses of mathematics from secondary school. The students in PBL Main may have a single course of mathematics, if enrolled in the Bachelor of Computer Science, or may have no mathematics beyond the middle of Secondary School, if from other programs.

In 2008, 325 students undertook the first offering of 1-unit PBL, with a weekly 1-hour lecture, supported with on-line forums and assessed through weekly assignments and a final examination of 1-hour duration. In 2009, 428 students were undertaking the 1-unit PBL course, with 102 students undertaking the 3-unit PBL course. The PBL-E course remained essentially the same in structure but the 3-unit course added an additional lecture per week and weekly tutorials. This allowed the development of the material in further depth. The majority of PBL Main homework was composed of two questions to be completed in the week, rather than the single question of PBL-E.



Lectures in PBL follow a set pattern. The first lecture of the week presents the solution to the previous homework, identifies the key points for this week's lectures and then builds on the topic area. The lecture concludes with the next assignment. Lectures are recorded and the lecture slides, recordings and all assignment work are available on the course electronic forum. The electronic forums also provide message boards for student interaction. PBL Main has a second lecture that develops the themes of the week's topic. Lecture materials are developed in parallel, with the single PBL-E lecture derived from a revision and abridgement of the two PBL Main lectures for that topic to maintain currency between the two courses.

Tutorials are offered for PBL Main and allow students to take part in collaborative problem solving exercises, with a tutor to provide assistance and guidance. Tutorial groups are up to 25 students, with sub-group formation of 5-8 students for problem solving. During these sessions, we introduce fundamental mathematical concepts that are useful in the later course, including counting and the bases of probability, including factorials, combinations and permutations. This addresses the difference in mathematical preparation that was identified in the intake. While a good grasp of mathematics can be useful for PBL, it is not essential. Problem specification has been of key concern to us, as the larger classes contain a percentage of students who are accustomed to a completely specified problem and are uncomfortable when confronted with problems that are not completely specified or, in the student's opinion, are not sufficiently and exactly specified. While some students regard this as a challenge, and also as an intellectual freedom, other students have found this to be a stumbling block.

Assessment of the course has proven to be one of the largest implementation issues. Students are interested in the material but their interest can easily be capped when they feel constrained by the assessment mechanisms, or feel that they haven't received sufficient, personalised, feedback. Early assessment for PBL-E revolved around a mark for each assignment out of 5, followed by feedback to the group demonstrating the marking scheme and the solution. The solution was also presented at the start of the lecture that corresponded to the hand-in time, to allow students to immediately gain feedback on the quality of their solution. PBL-E's student numbers posed a significant resource issue, as it takes approximately 2-3 minutes to mark each assignment, without detailed feedback. Thus, marking load starts at approximately 8 hours for each assignment and a team of markers have had to be employed and trained for consistency of response. PBL Main has a much smaller enrolment but employs detailed, personalised feedback, which also takes approximately 8 hours for a week's assignments. The requirement for a consistent and reproducible marking scheme that can be assigned to multiple markers constrains the range of problems that can be offered, as problems with too many possible solutions become effectively impossible to mark across 450 students. In response to this, we have considered many alternatives and are currently developing problems that may have multiple possible solutions, all of which may appear to be correct when, in fact, only one is. Again, this is an issue of problem and solution specification. Controlled use of multiple choice questions, with between 8 and 10 options, allows markers to quickly identify the flaw in reasoning and correctly mark the student. We are also investigating the possibility of reducing the dependency on a mark-based assessment for this course.

Early student response shows that students enjoy the course material and that it does develop their thinking skills, but a number of students, especially in PBL-E, encounter issues with the assessment model and perceived lack of feedback. For some students, their perception of the assessment mechanism can develop a very negative and unproductive approach to the course. We are actively seeking to address this, by allocating more resources to marking and feedback and through the use of automated marking mechanisms that allow more rapid response. Future implementations of PBL-E may include tutorials, or alternative assessment mechanisms.

## **First experiences: Carnegie Mellon University – a freshman seminar**

At Carnegie Mellon University, Pittsburgh, puzzle-based learning was offered as a 9 unit (3 credit) Freshman Seminar in spring 2009. Given the seminar nature of the course, enrollment was capped at

15 but it was encouraging to see that the wait list was longer than the class enrollment! The class had an inter-disciplinary mix of students majoring in Information Systems, Computer Science, Psychology, Statistics, Cognitive Science, Economics, and Physics. The class met twice a week for 80 minutes. In addition to what is described above in the Adelaide experience, given the smaller size of the class we were able to experiment with several alternative themes. For example, after the introductory classes, each session started with a puzzle-of-the-day. One student would present a puzzle of their choice. The class as a whole would try to solve the puzzle with hints and guidance provided by the puzzle poser. Student chosen puzzles ranged across the gamut of logic puzzles to diagrammatic reasoning to physical puzzles (tangrams). Students had to submit a one page write-up of their puzzle, solution, and most importantly, their reflection on the puzzle: what did they find interesting in the puzzle, variations, how does the solution tie into the general class discussions etc. In addition to the puzzle-of-the-day, we also conducted a *puzzlethon* where students again presented a puzzle of their choice. But this time the class voted on the best puzzle (a combination of presentation plus the nature of the puzzle) and prizes were distributed.

During our discussion of scientific induction and mathematical induction, given the smaller size of the class, we played Robert Abbott's inductive game of Eleusis (Gardner 1997) that models the process of scientific method. To introduce students to some of problem solving thoughts of leaders in the field we watched a few videos. These included Polya's "Let us teach guessing" wherein Polya beautifully illustrates several problem solving heuristics (that are embraced by puzzle-based learning) in the process of deriving a solution to the 5-plane problem; an interview with Herb Simon on being a researcher with advice to undergraduates; Richard Feynman on problem solving and induction; and Will Shortz's documentary "Wordplay" on crossword puzzles, their creators, and solvers. We also visited the Pittsburgh Super Computer Center open house to get a glimpse of problem solving in the real world. Student evaluation was done with components for class participation, puzzle presentations, homework assignments, three exams and a few in class quizzes. It has been gratifying to see that the response to the class has been favorable with some students commenting it was the best class they had that semester. A similar version of this course is being offered at Carnegie Mellon University in Doha, Qatar this summer. The freshman seminar is also slated to be offered in Pittsburgh in spring 2010.

## Conclusion

Puzzle-based learning is an experiment in progress. The goal is to foster general domain independent reasoning and critical thinking skills that can lay a foundation for problem-solving in future course work (as depicted in Figure 1). As fun as puzzles inherently are, they are just a means to this pedagogical end. Our preliminary experience in different contexts has been encouraging and well received as we continue to explore this approach.

## References

- Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist*, 26 (3 & 4), 369-398.
- Bransford, J. D., Sherwood, R. S., Vye, N. J., & Rieser, J. (1986). Teaching thinking and problem solving: Research foundations. *American Psychologist*, 41, 1078-1089.
- Danesi, M. (2002). *The Puzzle Instinct: The meaning of puzzles in human life*. Indiana University Press.
- Fisher, A. (2001). *Critical Thinking: An Introduction*, Cambridge, UK: Cambridge University Press.
- Gardner, M. (1961). *Entertaining Mathematical Puzzles*, New York: Dover Publications.
- Gardner, M (1997). *Penrose Tiles to Trapdoor ciphers*, Mathematical Association of America.

- Michalewicz, Z. & Michalewicz, M. (2008). *Puzzle-Based Learning: Introduction to Critical Thinking, Mathematics, and Problem Solving*, Melbourne: Hybrid Publishers ([www.PuzzleBasedLearning.edu.au](http://www.PuzzleBasedLearning.edu.au)).
- Parhami, B. (2009). Puzzling Problems in Computer Engineering, *IEEE Computer*, Vol.42, No.3, March 2009, pp. 26 – 29.
- Polya, G. (1945). *How to Solve It: A New Aspect of Mathematical Method*, Princeton: Princeton University Press.
- Poundstone, W. (2000). *How would you move Mount Fuji?: Microsoft's Cult of the Puzzle – How the World's Smartest Companies Select the Most Creative Thinkers*, Little Brown and Company.
- University of Adelaide: *Mathematics Working Party Final Report*, internal document, June 2008.
- Wing, J.M. (2006). Computational Thinking, *Communications of the ACM*, Vol.49, No.3, March 2006, pp.33 – 35.

## Biographical Information

Nickolas Falkner is an Associate Lecturer at the School of Computer Science, The University of Adelaide. Nick coordinates the School's assessment programs and is responsible for the coordination of a number of core programs, with a focus on innovative and effective teaching practices. Prior to joining the University, he held teaching and management positions in network engineering and systems management. He received his PhD in Computer Science from The University of Adelaide. He carries out research in wireless sensor networks, data stream management and semantic consistency in large systems. Contact him at the School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia;

[jnick@cs.adelaide.edu.au](mailto:jnick@cs.adelaide.edu.au); [www.cs.adelaide.edu.au/~jnick](http://www.cs.adelaide.edu.au/~jnick)

Zbigniew Michalewicz is a Professor at the School of Computer Science, The University of Adelaide. He also holds professorships at the Institute of Computer Science, Polish Academy of Sciences, and the Polish-Japanese Institute of Information Technology. He chairs the board of SolveIT Software, a technology company he co-founded, and holds an honorary professorship at the State Key Laboratory of Software Engineering of Wuhan University, China. His current research interests are in evolutionary computation. Professor Michalewicz received a PhD in Applied Mathematics from the Institute of Computer Science, Polish Academy of Sciences. Contact him at the School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia;

[zbyszek@cs.adelaide.edu.au](mailto:zbyszek@cs.adelaide.edu.au); [www.cs.adelaide.edu.au/~zbyszek](http://www.cs.adelaide.edu.au/~zbyszek)

Raja Sooriamurthi is an Associate Teaching Professor of Information Systems at Carnegie Mellon University, Pittsburgh. Prior to joining Carnegie Mellon, he held academic positions at the Kelley School of Business, Indiana University and the Computer Science Department of the University of West Florida. He has a PhD in computer science from Indiana University. His research interests are in the broad area of artificial intelligence and case-based reasoning. He is the current president of Alpha Iota Delta, the international honor society for information systems and decision sciences. A passionate instructor, he has won several awards for distinguished teaching. Contact him at the Information Systems Program, Carnegie Mellon University, Pittsburgh, PA 15213;

[raja@cmu.edu](mailto:raja@cmu.edu); [www.andrew.cmu.edu/~sraja](http://www.andrew.cmu.edu/~sraja)