Puzzle-Based Learning for Engineering and Computer Science

Nickolas Falkner, *University of Adelaide*
Raja Sooriamurthi, *Carnegie Mellon University*
Zbigniew Michalewicz, *University of Adelaide*

To attract, motivate, and retain students and increase their mathematical awareness and problem-solving skills, universities are introducing courses or seminars that explore puzzle-based learning. We introduce and define this learning approach with a sample syllabus and course material, describe course variations, and highlight early student feedback.

A recent article describes a puzzle-based freshman seminar introduced at the University of California, Santa Barbara, to motivate and retain computer engineering students.[1] The author argues that attracting students to computer science and engineering programs represents only one aspect of a broader problem, 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 more skilled graduates capable of solving real problems of innovation in a changing environment. Missing in the majority of engineering and computer science curricula is a focus on developing problem-solving skills. Further, many courses that introduce elements of problem-solving skills do so at the programs' third or fourth level, after students have already faced the majority of their in-academy intellectual challenges.

While some courses with a design content emphasis might meet this requirement, 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 end of each chapter, solved using material discussed 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 a guidebook. One of our favorite examples for illustrating 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?*

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 you've learned?

Students often have difficulty applying 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[2] wrote "though many teachers would claim to teach their students 'how to think', most

Published by the IEEE Computer Society

0018-9162/10/$26.00 © 2010 IEEE

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

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 students' decreasing mathematical skills. A recent Mathematics Working Party Final Report, issued by the University of Adelaide,[3] 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 puzzle-based learning course to address all the issues raised here.

PUZZLE-BASED LEARNING APPROACH

The puzzle-based learning approach aims to encourage engineering and computer science students to think about how they frame and solve problems not encountered at the end of some textbook chapter. Our goal is to motivate students while increasing their mathematical awareness and problem-solving skills by discussing a variety of puzzles and their solution strategies. The course is based on the best traditions introduced by Gyorgy Polya and Martin Gardner over the past 60 years.[4,5]

Many teachers have used puzzles for teaching purposes, and the puzzle-based learning approach has a tradition that exceeds 60 years.[6] Historians found the first mathematical puzzles in Sumerian texts from circa 2500 BC. However, some of the best evidence for the puzzle-based learning approach can be found in the works of Alcuin, an English scholar born around 732 AD, whose main work, *Problems to Sharpen the Young*, included more than 50 puzzles. Some 1,200 years later, one of Alcuin's puzzles—the "river crossing problem"—is still used in artificial intelligence textbooks to educate computer science students.

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

Independence

The puzzles are not specifically tied to a single problem-solving domain.

Generality

Educational puzzles should explain some universal mathematical problem-solving principles. This is key. Most people agree that problem solving can only be learned by actually solving problems. This activity, however, must be supported by instructor-provided strategies. These general strategies allow for solving yet unknown problems in the future.

Simplicity

Educational puzzles should be easy to state and remember. This is important because easy-to-remember puzzles increase the chance students will remember the solution method, including the universal mathematical problem-solving principles.

Eureka factor

Educational puzzles should initially frustrate the problem solver, but hold out the promise of resolution. A puzzle should be interesting because its result is not immediately intuitive. Problem solvers often use intuition to start their quest for the solution, and this approach can lead them astray. Eventually they reach a "Eureka moment"—Martin Gardner's Aha!—when students recognize the correct path to solving the puzzle. A sense of relief accompanies this moment, and the frustration felt during the process dissipates, giving the problem solvers 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.

There is a strong connection between the ability to solve puzzles and the ability to solve industry and business problems.

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 such as playing in a casino environment, fighting against dragons, or dropping eggs from a tower.

THE LURE OF EDUCATION

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 that helps retain and motivate students. Above all, they are responsible for developing 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 and business problems. Many real-world problems can be perceived as large-scale puzzles. William Poundstone, when investigating the purpose of famous Microsoft/Silicon Valley interview puzzles,[7] wrote,

> "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

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

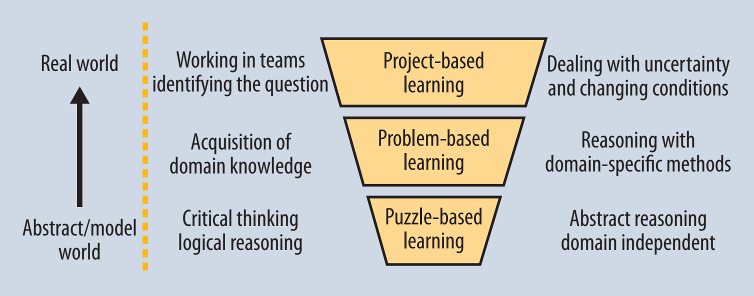

Figure 1. Problem solving in the real world requires a continuum of learning and skills in which each layer of skills builds upon the layers below it.

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 versus problem-based versus 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; and critical thinking and applying general problem-solving strategies.

These three skill categories are captured in the three forms of learning Figure 1 depicts. In this continuum, each layer of skills builds upon the layers below it. Puzzle-based learning focuses 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 and project-based learning are well established methodologies.[8,9] By our description, problem-based learning requires significant domain knowledge. This is the form of learning typically emphasized in a domain-specific undergraduate course such as electromagnetism, data structures, or circuit theory. Project-based learning, on the other hand, deals with complex situations in which usually no clearly unique or correct way of proceeding exists. For example, "How can we increase the adherence of cystic fibrosis patients to following their treatment protocol?" Determining the best solution in such a situation can be difficult.

The pedagogical objectives of project-based learning include dealing with ambiguity and complexity, integrating a variety of approaches, user testing of the proposed solutions' value, and working with a team composed of diverse backgrounds and skills. In both problem-based and project-based learning, the problem drives the learning: Students must assess what they already know, the knowledge they need to address the problem, and how to bridge the knowledge and skill gap.

Puzzle-based learning focuses on domain-independent critical thinking and abstract reasoning. This leads us to ask, "What is the difference between a puzzle and a problem?" One way to characterize the difference measures the extent to which domain-specific knowledge is needed to solve it. The general flavor of puzzles asserts that their solution should require only domain-neutral general reasoning skills—biologists, musicians, and artists should all be able to solve the same puzzle. The different styles of reasoning required for problem-based and puzzle-based learning could be compared to the difference between a field investigator and an armchair detective: one emphasizes pure reasoning more.

Dropping eggs

The well-known egg-drop experiment provides an example that compares and contrasts problem-based and puzzle-based learning. The traditional problem-based learning version of this experiment involves finding a way to drop an egg from a maximal height without breaking it. A puzzle-based learning version of this experiment also involves dropping an egg from a building, but the question under investigation, although related, is quite different.

In the problem-based learning approach, students conduct a series of physical experiments to determine how to maximize the height from which an egg can be dropped without breaking. There are two broad approaches: Dampen the impact's effect (leading to padding-based solutions) or lessening the impact (leading to delivery mechanisms such as a parachute). The team-based learning outcomes of such an experiment determine different ways to dampen or lessen an impact.

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 the following: Using multiple eggs, what would be an effective strategy for determining the highest floor from which I could drop an egg without breaking it? This question has interesting variations. This thought experiment has three entities: the number of eggs, number of drops, and 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 $F_{e,d}$.

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

An alternate question could be, "Given a fixed number of floors—say, 100—and a number of eggs—say, three—what is the maximum number of drops needed to determine the breaking floor ($D_{e,f}$)?" Yet another version might 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 goal 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 include "all eggs behave the same way" and "an egg that survives a drop is not damaged 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."

Suppose, 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 ($e = 1$). 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 ten-floor building by starting at floor one and working our way up. Now suppose we had two eggs ($e = 2$). 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 encouraged to contribute and refine their suggestions, with controlled prompting. By considering examples and reasoning about what happens if the first egg does or 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 pedagogical goals with the emerging paradigm of computational thinking.[10] Puzzle-based learning resonates with the computational thinking emphasis on abstraction and analytical thinking. With reference to Figure 1, computational thinking straddles the 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 computational thinking in the curriculum.

PUZZLE-BASED LEARNING COURSES

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

One important point about puzzle-based learning courses is that they are 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 illustrations of the concepts presented. 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.

The process of understanding problem-solving principles leads students through a variety of topics, exposing them to many important concepts.

Despite a variety of possible puzzle-based learning offerings, the course's structure is very much the same. The topics listed below correspond to a 12-week semester regardless of whether each topic is allocated one hour or three. Although the topics have some dependency, as we build and develop our model for problem solving, the later topics' order can be rearranged. The topic structure also supports a high-level first pass and a secondary, detailed pass model for younger students or for development over a multiyear curriculum. Similar topic structures are employed across both secondary and tertiary student environments, as well as our presentations of puzzle-based learning concepts to industry, as the following list shows:

1. *Introduction.* What it is it all about?
2. *The problem.* Rule #1: Understand the problem.
3. *Intuition.* Rule #2: Be cautious of your intuitions—guess, but verify.
4. *Modeling.* Rule #3: Reason, model, calculate.
5. *Some mathematical principles.* Do you know how much you already know?
6. *Constraints.* Am I aware of all of the constraints?
7. *Optimization.* What is the best arrangement, and is it one that I can actually use?
8. *Probability.* Counting skills and counterintuitive results.
9. *Statistically speaking.* What does an apparently convincing statement actually mean?

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

10. *Let's simulate.* Can we generate the answer?
11. *Pattern recognition.* What is next?
12. *Strategy.* Shall we play?

We illustrate each topic with a variety of puzzles presented interactively. The course introduces a few simple problem-solving rules that we refer to in every class. Each week, students receive homework assignments that cover one or more puzzles addressed in class. The following week, at the beginning of class, the instructor presents and discusses solutions. In one instance of the course, homework contributes 30 percent toward the final grade, and the final exam contributes the remaining 70 percent. Students can access all lecture slides, audio lecture recordings, and additional material, including course software. Sample course work might include a sample lecture that poses the following question: "The problem: What are you after?" The lecture introduces the most important problem-solving rule.

Without understanding the problem, all efforts to find a solution usually simply waste time: the dictum of *solve the right problem and solve the problem right.*

First rule of problem solving

Be sure you understand the problem and all the basic terms and expressions used to define it.

Indeed, without understanding the problem, all efforts to find a solution usually simply waste time: the dictum of *solve the right problem and solve the problem right*. Underspecification can be used as a tool that encourages students to determine what they know, what they don't know and must find out, and what they cannot find out.

The approach places the emphasis for knowledge discovery on the students and forces them to accept that, on occasion, they must provide their best guess. The first puzzle we use to illustrate this simple observation is one of Martin Gardner's favorites.

Puzzle 1

A farmer has the following: 20 pigs, 40 cows, and 60 horses. How many horses does he have, if he calls the cows horses?

It takes students a short time to understand the problem, "calling," which has little to do with "having." The farmer still *has* 60 horses.

This example can be followed by another classic.

Puzzle 2

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

Again, many students would jump immediately into the obvious answer of 50 kph without much understanding of what the average speed is—or rather, how average speed is defined. Most students are surprised to discover the correct answer of 48 kph. The next time, in any course they take in their programs, they will think twice before they answer a question on some average. We seek to foster this clear and thoughtful analysis to hone and guide intuition.

Homework

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

One weekly assignment given to the students at the end of this lecture proceeded as follows.

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 (when sand in the 11-minute hourglass stops), 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 (when the sand in an 11-minute hourglass stops), 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 dropped into the water.

Which of these solutions is simpler? Solution A takes 15 minutes to complete and requires two inversions.

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

Solution B requires 22 minutes, but only one inversion. Most students believed that solution A was simpler, as it required less time. They were, however, 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, but also on what the simplest solution means. This also introduces the concept of multiobjective optimization, a concept students are usually exposed to during their third year of studies.

OTHER EXAMPLES

We can use many different puzzles to illustrate fundamental concepts of probability, statistics, pattern recognition, games, constraint processing, and optimization. Here we present a few examples.

Puzzle 3

A farmer sells 100 kg of mushrooms for $1 per kg. The mushrooms contain 99 percent moisture. A buyer makes an offer for these mushrooms at the same price a week later. However, another week later the mushrooms would have lost 98 percent of their moisture content. How much will the farmer lose if he accepts the offer?

This is a good example for resisting immediately intuitive answers—it might not be 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 shaken, and a ball is then randomly removed. This ball happens to be white. What is the probability now that the bag currently contains the white ball? Puzzle 4 thus introduces basic concepts in probability.

Puzzle 5

There are n dots on the flat surface of a plane. Two players, A and B, move alternatively, with A moving first. The game's rules are the same for both players: At each move, they can connect two points, but they cannot connect points that were already directly connected to each other or connect a point with itself. They build a graph with predefined n vertices by connecting some of the dots.

The winner is the one who connects the graph (a graph is connected if there is a path between any two nodes of the graph; however, not every two nodes must be connected directly). Which player, A or B, has a winning strategy?

This puzzle introduces graphs and investigating the concept of strategies—discovering the winning strategies of the first or second player is *not* trivial.

More puzzles

We collected and organized a few hundred educational puzzles 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 students' progress 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 encountering a final exam loaded with puzzles. However, the exam can be organized in many ways that will make it meaningful. For example, last semester the final exam questions included the following two examples.

From this point, students could still discuss different interpretations of the result, encouraging thought.

Question 1

Five minutes after midnight of April 13th, 2004, a heavy rain fell in Melbourne. What is the probability that, 72 hours later, it would be sunny there? Justify your answer.

This question checked students' skills for understanding the problem before providing an answer. (For the curious, the probability is zero percent.)

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 tested students' modeling skills and rewarded them for identifying the implicit question "When should the hands of a watch cover each other," and for modeling the problem, even if they reached an incorrect conclusion.

Wide range

As a glimpse into the range of approaches to a problem consider the following: In class we examined different reasoning styles—quantitative versus qualitative versus intuitive versus wild guess. Both quantitative and qualitative methods are rigorous, using numbers and algebra while the other uses language and logic. One student group went through the precise calculations to determine when an overlap should occur on a correct watch, while another group qualitatively

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

reasoned that if a clock were correctly running, then at 60 minutes past noon, the minute hand would be over the 12 and the hour hand over the 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, we direct readers to the website associated with the text, www.PuzzleBasedLearning.edu.au.

UNIVERSITY OF ADELAIDE

We now explore two of this course's implementations: the primary development site at the University of Adelaide and another at Carnegie Mellon University, Pittsburgh. The initial implementation of puzzle-based learning offered a one-unit course set as a component of a thee-unit first-year course, *Introduction to Electronic Engineering*. After gaining support from the Dean of the Faculty of Engineering, Computer and Mathematical Sciences, from 2009, the university placed this one-unit offering into introductory courses across the engineering programs available within the university.

Students are interested in the material, but their interest can easily be capped when they feel constrained by the assessment mechanisms.

A three-unit, first-year course for students planning to major in computer science launched simultaneously in 2009, and was made available to all nonengineering students at the university. We refer to the one-unit offering as PBL-E (PBL for Engineers), and the three-unit offering as PBL Main. The courses cover the same material, but 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 have also taken two mathematics courses from secondary school. The students in PBL Main might have a single course of mathematics if enrolled in the Bachelor of Computer Science program, or might have no mathematics beyond the middle of Secondary School, if from other programs.

In 2008, 325 students undertook the first offering of one-unit PBL, with a weekly one-hour lecture, supported with online forums and assessed through weekly assignments and a final, hour-long examination. In 2009, 428 students undertook the one-unit PBL course, with another 102 students undertaking the three-unit PBL Main course. The PBL-E course remained essentially the same in structure, but the three-unit course added an additional lecture per week, along with weekly tutorials. This let instructors explore the material's development in further depth. The majority of PBL Main homework consisted of two questions to be completed during the week, rather than the single question posed by PBL-E.

Patterns of learning

Lectures in PBL follow a set pattern. Each week, the first lecture presents the solution to the previous homework, identifies key points for that week's lectures, 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 made available on the course's electronic forum. These 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.

The university offers tutorials for PBL Main that let students take part in collaborative problem-solving exercises, while a tutor provides assistance and guidance. Tutorial groups can hold up to 25 students, divided into subgroups of five to eight for problem solving. During these sessions, we introduce fundamental mathematical concepts useful in the later course, including counting and the bases of probability, such as factorials, combinations, and permutations. This addresses the differences in mathematical preparation identified in the intake.

While a good grasp of mathematics can be useful for PBL, it is not essential. Problem specification has been a key concern, as the larger classes contain students accustomed to a completely specified problem, and thus feel uncomfortable when confronted with problems not completely specified or, in the student's opinion, not sufficiently and exactly specified. While some students regard this as a challenge, and also as an intellectual freedom, others have found it 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 they haven't received sufficient, personalized feedback. Early assessment for PBL-E revolved around a mark for each assignment out of five, followed by feedback to the group that demonstrated the marking scheme and solution. Instructors also presented the previous lesson's solution at the lecture's start, which corresponded with the

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

hand-in time, to let students immediately gain feedback on the quality of their solution.

PBL-E's student numbers posed a significant resource issue: Without detailed feedback, it takes approximately two to three minutes to mark each assignment. Thus, the marking load starts at approximately eight hours for each assignment, with a team of markers employed and trained to provide consistency of response.

PBL Main has a much smaller enrollment but employs detailed, personalized feedback that also takes approximately eight hours to complete 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. 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 might 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 eight and ten options, lets markers quickly identify the flaw in reasoning and correctly mark the student's work. We also investigate the possibility of reducing the dependency on a mark-based assessment for this course.

Early student response shows that they enjoy the course material and it does develop their thinking skills. However, several students, especially in PBL-E, encounter issues with the assessment model and its perceived lack of feedback. Others worry that the assessment mechanism can develop a negative and unproductive approach to the course. We are actively seeking to address these concerns 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.

CARNEGIE MELLON UNIVERSITY

Carnegie Mellon University offered puzzle-based learning as a nine-unit, three-credit freshman seminar in spring 2009. Given the spring course's seminar nature, instructors capped enrollment at 15, but we found it encouraging to see that the wait list exceeded the class's enrollment. The class had an interdisciplinary 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 the knowledge gained from the Adelaide experience, the smaller size of this class let us experiment with several alternative themes. For example, after the introductory classes, each session started with a puzzle of the day.

First, a student would present a puzzle of his or her choice. The class as a whole tried to solve this puzzle, with hints and guidance provided by the puzzle poser. Puzzles chosen by the students ranged the gamut from logic puzzles to diagrammatic reasoning to physical-puzzle tangrams. Students submitted a one-page write-up of their puzzle, solution, and—most importantly—their reflection on the puzzle: What captured their interest in the puzzle and its variations, and how did the solution tie in with the overall course's instruction?

In addition to the daily puzzle, we also conducted a puzzlethon that again presented a puzzle of the students' choice. But this time the class voted on the best puzzle (a combination of presentation and the puzzle's nature) and instructors distributed prizes to the winners.

During our discussion of scientific and mathematical induction, given the smaller size of the class, we played Robert Abbott's inductive game *Eleusis*,[11] which models the scientific method. To introduce students to some problem-solving thoughts of leaders in the field, we watched a few videos. These included Polya's "Let us teach guessing" in which he beautifully illustrates several problem-solving heuristics embraced by puzzle-based learning in the process of deriving a solution to the five-plane problem; an interview with Herb Simon on being a researcher with advice to undergraduates; Richard Feynman's discussion of problem solving and induction; and Will Shortz's documentary *Wordplay* on crossword puzzles, their creators, and solvers.

To emphasize the link between the thought processes involved in solving puzzles and addressing open real-world problems, we examined a few case studies.

We also visited the Pittsburgh Supercomputing Center open house to glimpse problem solving in the real world. To emphasize the link between the thought processes involved in solving puzzles and addressing open real-world problems, we examined a few case studies, including the recently cracked Netflix Prize (www.netflixprize.com) and the classic work of Mosteller[12] in resolving the authorship of some of the disputed Federalist papers.

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. Some have also commented on the revisiting of PBL themes in their subsequent

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.

discipline-specific courses. Carnegie Mellon University's Doha, Qatar, campus hosted a summer 2009 version of this course, while the freshman seminar is slated to be offered in Pittsburgh during spring 2010.

Puzzle-based learning is an in-progress experiment that seeks to foster general domain-independent reasoning and critical thinking skills that can lay a foundation for problem-solving in future course work. As fun as puzzles inherently are, they provide only a means to this pedagogical end. Our preliminary experience in different instantiations of the course and educational contexts has been encouraging and well received as we explore this approach. We continue collecting relevant data to demonstrate the benefit of our approach. Early results[13] indicate that students who enroll in our course perceive an improvement in their thinking and general problem-solving skills. ■

References

1. B. Parhami, Puzzling Problems in Computer Engineering, *Computer*, Mar. 2009, pp. 26-29.
2. A. Fisher, *Critical Thinking: An Introduction*, Cambridge University Press, 2001.
3. University of Adelaide, *Mathematics Working Party Final Report*, internal document, June 2008.
4. G. Polya, *How to Solve It: A New Aspect of Mathematical Method*, Princeton Univ. Press, 1945.
5. M. Gardner, *Entertaining Mathematical Puzzles*, Dover Publications, 1961.
6. M. Danesi, *The Puzzle Instinct: The Meaning of Puzzles in Human Life*, Indiana Univ. Press, 2002.
7. W. Poundstone, "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, 2000.
8. P.C. Blumenfeld et al., "Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning," *Educational Psychologist*, vol. 26, nos. 3 and 4, 1991, pp. 369-398.
9. J.D. Bransford et al., "Teaching Thinking and Problem Solving: Research Foundations," *Am. Psychologist*, vol. 41, 1986, pp. 1078-1089.
10. J.M. Wing, "Computational Thinking," *Comm. ACM*, Mar. 2006, pp. 33-35.
11. M. Gardner, *Penrose Tiles to Trapdoor Ciphers*, Mathematical Assoc. of America, 1997.
12. F. Mosteller and D. Wallace, "Deciding Authorship," *Mathematics: People, Problems, Results*, D.M. Campbell and J.C. Higgins, eds., Wadsworth Publishing, 1984.
13. N.J.G. Falkner, R. Sooriamurthi, and Z. Michalewicz, "Puzzle-Based Learning: The First Experiences," *Proc. 20th Ann. Conf. Australasian Assoc. for Eng. Education* (AaeE 09), 2009.

Nickolas Falkner *is a lecturer in the School of Computer Science at the University of Adelaide. He received a PhD in discovery and classification of information in large systems from the University of Adelaide. His research interests include wireless sensor networks, automated network configuration, data fusion, and data stream management. He is also active in educational research, with a focus on increasing student participation, retention, and enthusiasm. Contact him at jnick@cs.adelaide.edu.au.*

Raja Sooriamurthi *is an associate teaching professor with Carnegie Mellon's Information Systems Program and also has a faculty appointment at Carnegie Mellon's Heinz College. He received his PhD in computer science from Indiana University, Bloomington. His research interests are in artificial intelligence, case-based reasoning, and CS/IS pedagogy. His pedagogical efforts have been recognized with several awards for distinguished teaching. Sooriamurthi is a member of the IEEE Computer Society, the ACM, and Decision Sciences Institute. Contact him at raja@cmu.edu.*

Zbigniew Michalewicz *is a professor in the School of Computer Science, University of Adelaide. He is also at the Institute of Computer Science, Polish Academy of Sciences and at the Polish-Japanese Institute of Information Technology, Poland. He received a PhD in computer science from the Polish Academy of Science. His research interests include problem-solving methodologies, evolutionary computation, modern heuristic methods, and adaptive business intelligence. Michalewicz is a Fellow of the Australian Computer Society. Contact him at zbyszek@cs.adelaide.edu.au.*

cn Selected CS articles and columns are available for free at http://ComputingNow.computer.org.

For more information on any topic presented in *Computer*, visit the IEEE Computer Society Digital Library at

www.computer.org/csdl

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on April 12,2010 at 16:39:46 UTC from IEEE Xplore. Restrictions apply.