Int. J. Engng Ed. Vol. 19, No. 6, pp. 810–817, 2003
Printed in Great Britain.

0949-149X/91 $3.00+0.00

The Common Skills of Problem Solving: From Program Development to Engineering Design*

SHARI J. KIMMEL
Penn State Berks-Lehigh Valley College and Lehigh University, Pennsylvania, USA

HOWARD S. KIMMEL and FADI P. DEEK
New Jersey Institute of Technology, Newark, NJ 07102, USA. E-mail: deek@njit.edu

The introduction of engineering design in the first year of the curriculum has become commonplace in order to provide students with early experiences in engineering principles and exposure to real-world applications. Many different approaches to the development and implementation of these courses are used but regardless of the method or specific emphasis, students enrolled in engineering design classes are expected to be problem solvers and to communicate effectively, both verbal and written. We have adapted and integrated a problem-solving and program development methodology originally used in a computer science environment to an introductory engineering design class which helps beginning engineering students develop these important skills. We have also conducted a base-line study in this engineering design course to evaluate this methodology and its impact on students' problem-solving abilities, skills, knowledge, and attitudes in a first-year course on engineering design.

INTRODUCTION

AS THE ABET Engineering Criteria 2000 is implemented, engineering schools will be re-examining the skills and knowledge that students are acquiring during their course of instruction [1]. Problem solving is one skill that will need to receive increased attention as the EC2000 criteria are attended to. Unfortunately, many students enter undergraduate engineering programs lacking basic problem-solving skills. Prior life experiences certainly involve solving routine problems and can be the basis for further learning. But solving such problems is usually accomplished in an intuitive fashion. Intuition alone is not sufficient within the academic context and systematic strategies and heuristics are called for to solve more complex problems. Typically, students' method of solving problems involves rote memorization of the material in order to get through the exams, which often consist of simple, well-defined problems, then forgetting it almost immediately thereafter. Furthermore, many students would not be able to solve problems that involved changes in wording or context. Though engineering design textbooks often do discuss the problem-solving aspect of engineering design, they usually provide either an abstract list of steps that are difficult to directly implement or too simplistic a solution process which applies only to already well defined problems [2].

For engineers, the term 'design process' can be used almost interchangeably with 'problem-solving process'. Design is essentially a logical, sequential process intended to solve engineering problems. Problem solving and design requires critical thinking skills, something that students tend to view as hard work and something to avoid [3]. Schools of Engineering have developed freshman engineering design courses to provide the first-year students with early experiences in engineering, and students enrolled in these courses are expected to be problem solvers. While the characteristics and objectives of such courses may vary, almost all of them include objectives that emphasize the ability to set up and solve problems, and the ability to communicate effectively, both verbal and written [4].

Unfortunately, the typical course in freshman engineering design provides similar superficial treatment for the students. Certainly, the reasons include that problem solving skills are perceived to be somewhat vague and difficult to objectively assess and that most faculty probably do not know how or feel comfortable teaching it. 'Problem solving' is a phrase that might be found in a course description, but unless it is included in the course objectives and specifically identified as a skill students are expected to master in the course, it is, at best, given perfunctory attention in the classroom [5]. It has been demonstrated that true mastery of skills comes only from practice in the application of the skills in real-world situations [5]. In addition, time is needed for students to learn and master these skills. A stronger emphasis in

* Accepted 28 March 2003.

problem solving and design will have to be evident in these courses if EC2000 is to be met.

We have adapted the problem solving heuristics originally used in a computer science environment to an introductory engineering class to help freshman engineering students develop these skills. The introductory Engineering Design and Graphics course (ED&G 100) at Penn State—Berks Campus exposes students to conventional drafting techniques, computer graphics, solid modeling and engineering design. The typical class consists of mostly first-year and some second-year students with a wide range of skills and experience.

During the fall 2000 semester, a section of ED&G 100 with 20 students taught by the first author included writing and problem-solving exercises integrated into the curriculum. The major assignment in this course is a group design project in which students apply skills previously learned in the class. In the fall 2000 semester, students were presented with a cart used in an industrial setting but which the company wanted redesigned to be safer and easier for the workers to use and more space efficient than the original at a reasonable cost.

PROBLEM-SOLVING METHODOLOGY AND APPLICATION TO ENGINEERING DESIGN

Computing and engineering share much common ground; problem solving is a fundamental skill necessary to succeed in both disciplines. Before we discuss problem solving in engineering design, we provide a concise introduction to the general problem-solving method.

A problem is a matter that requires creative and logical thought; the method used in solving the problem is the 'process' that assists in eventually producing a 'product', or a solution [6–8]. A variety of views about thinking and problem solving has been proposed. Notable among them are: behavioral [9], information processing [10], and constructivism [11]:

- The behavioral approach views human problem solving '. . .as a stimulus (input) and a response (output) without speculating about the intervening process' [12].
- The information processing approach to human problem solving is defined as '. . . a form of means-end analysis that aims at discovering a process description of the path that leads to a desired goal . . .' [10]. Means-end analysis is a widely used strategy where the problem solver defines a problem, identifies a desired goal, asks questions, uncovers obstacles, and specifies operators to surmount these obstacles and reach a goal [8, 10, 13].
- The constructivist approach, grounded in Piaget's work, describes problem solving as an evolving knowledge-construction and understanding process utilizing past experiences, resources, as well as active assimilation of new information.

Both the constructivist and information-processing approaches are branches of the cognitive perspective in which skills such as reasoning, comprehending, and planning are central concepts in problem solving [14, 15].

Regardless of the problem-solving approach, the process consists of a series of independent stages that may take place either in sequence or in parallel. Using a systematic approach to solve problems is of particular importance. Many methods of problem solving have been produced.

One popular approach was proposed by the mathematician George Polya [16], who defined a four-stage process for solving a problem:

- understanding the problem
- devising a plan
- carrying out the plan
- looking back.

Each stage focuses on a unique aspect of the problem. The first step is concerned with understanding the problem's question and requirements. The comprehension of the problem requires the identification of the goal, the givens, the unknowns, the conditions, the constraints, and their interdependence. Devising the plan is the outline and refinement of a potential solution to the problem. Carrying out the plan is the transformation of the plan into a concrete reality and producing a solution to the problem. Finally, looking back is the confirmation of the result and the assessment of correctness of the solution.

No perfect method has been identified to solve problems, and problems may still be solved incorrectly even with the most structured technique. Mathematical problem solving relates conditions to conclusions by means of proofs: 'given a set of conditions, the proof of a theorem exhibits the truth of its statement, subject to the conditions' [17]. Algorithmic problem solving, on the other hand, uses techniques, such as structured decomposition and stepwise refinement [18–20], and facts, such as givens and unknowns, to outline steps leading to a problem's solution.

Most problem-solving methods are generic and can be applied to a variety of domains, though some were developed to provide mathematics or science students with an explicit method for solving problems. Deek [20] developed a domain-specific model to be used in computer programming. This model was adapted to fit the needs of students involved in engineering design, which is discussed in this paper. We believe that the problem solving and program development model, used in freshman level computing courses at NJIT for the past few years [21–22], is complementary to the engineering design pedagogy. Computer science instructors teach their students to develop the algorithms that are necessary to create useful and usable solutions and programs. These skills are also what engineering instructors want their students to know. To develop useful algorithms, students should realize that problem solving is a recursive process, where information obtained and skills gained at any one point in the process may, and often must, be revisited at other points along the way to a solution. For example, the potential solutions should be reviewed to see that it satisfies the objectives and revise the potential solutions if they don't.

Next, we discuss the tasks involved in each of the five stages of the problem-solving process for engineering design: formulating the problem, planning the solution, designing the solution, testing the solution, and delivering the solution. A graphical description of the process is shown in Fig. 1. This model takes into consideration the activities, knowledge and skills required to solve engineering design problems:

- *Problem formulation* requires the construction of a well-defined description through refinement of the given problem statement, including diagrams, mathematical formulation, etc. The preliminary problem understanding is obtained by refining the problem description using inquiry questions. A structured representation is obtained by extracting and organizing the relevant information (goal, givens, unknowns, conditions, constraints) from the problem description.
- *Solution planning* requires a strategy to be discovered by first identifying alternative solutions using heuristics such as solving simpler, related or analogous problems. Problem decomposition follows where the original problem is decomposed into a collection of intermediate subproblems based on the strategy selected; these are then decomposed into sub-subproblems, and so on. Relevant data organization, based on problem decomposition, is accomplished by associating with each subproblem its givens and unknowns.
- The next stage, *solution design*, requires the high-level plan produced by the proceeding planning stage be refined. This involves the sequencing of subproblems, the determination of whether the subproblems require further decomposition, and the establishment of hierarchical relationships among the various solution components. The sub-components are now viewed as units whose functions must be specified and the data associated with these units (input, output, and intermediate data) are more formally represented. A final detailed design task transforms each subproblem into a corresponding solution specification.
- The *solution testing* stage requires generating test cases on the basis of problem requirements and certifying that the proposed solution satisfies the test results. Test results must be verified not only for correctness and completeness but also for performance criteria such as reliability, usability, etc. Modification to the design, strategy or even problem formulation may be required on the basis of testing. In addition to product testing, process evaluation and feedback is an integral part of the problem solving methods.
- The last stage, *solution delivery*, requires that information produced during the course of previous stages be organized and presented. The documentation of the solution strategy, design, and test results is important for subsequent refinements and updates.

We have developed and implemented a base-line study in an engineering design course to evaluate this methodology and its impact on students' problem solving abilities, skills, knowledge, and attitudes in a first-year course on engineering design.

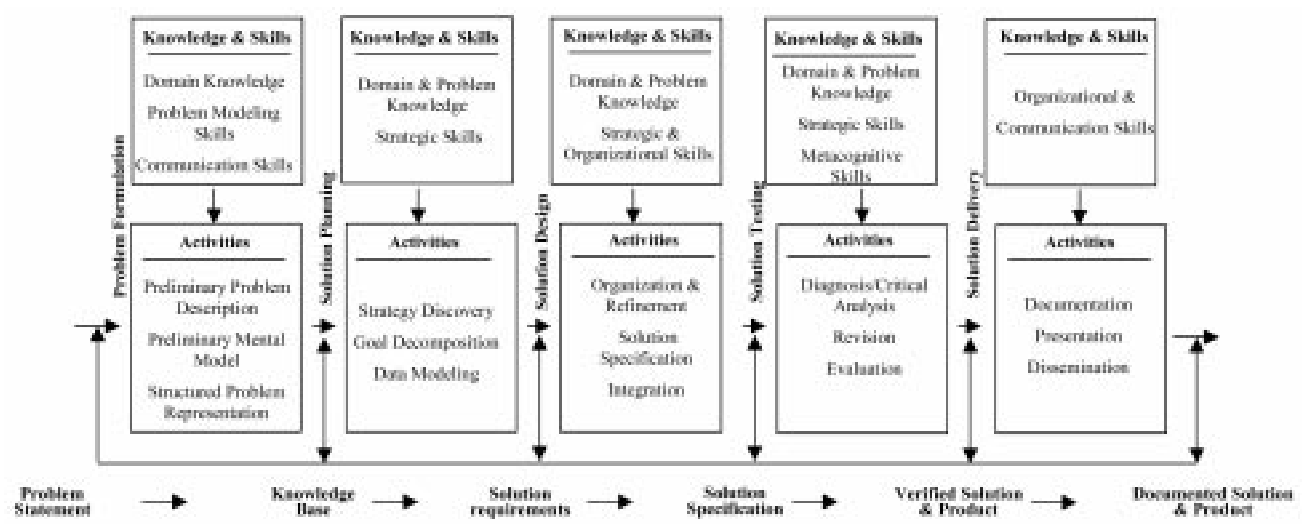

Fig. 1. The activities, skills and knowledge of problem solving in engineering design.

IMPLEMENTATION

The process of engineering design can naturally benefit from the inclusion of problem-solving activities. The use of a problem-solving heuristic, such as that described in the previous section, can help students learn how to tackle difficult problems on their own. In the ED&G 100 course, assignments specifically designed to enhance students' problem-solving skills were introduced within the context of a design project. Assignments related to the design project tried to emphasize the importance of the preliminary problem description and to teach students how to create a structured problem representation for the problem formulation step. The solution planning and solution design steps were implemented through brainstorming in the design component and through encouraging the students to write out solution steps. The solution testing aspect requires students to perform calculations on their individual designs and to implement their solution. The solution delivery step involves producing professional looking reports, both individually and in groups, and giving oral presentations. Allocating enough time to give students adequate practice in all these steps in a one semester course is generally difficult, especially since problem-solving skills are often given a back seat to other course objectives.

The students were given the opportunity to apply all the steps of the problem-solving process to a problem that is not well defined. Initially, the students were asked to define the problem given unstructured assignments. The results of these assignments were largely unsuccessful. Many students did not give responses which demonstrated a proper understanding of the problem, either because they did not read the problem statement carefully enough or they did not put in the effort to consider the ideas presented in the problem statement. Though students did, to some extent, apply the entire problem-solving heuristic to the design project, they required significant guidance through each of the steps to successfully understand and implement each one. Since this study describes a first attempt to implement this problem-solving heuristic in an engineering design class, the amount of guidance required by students was not anticipated in advance.

The need to focus on the first step of the heuristic; i.e., the problem formulation, was apparent. Previous studies have demonstrated this step as a major obstacle to successful problem solving by students. Stuart [23] found that few students understood how to formulate or re-describe the problem. Woods *et al.* [24] carried out a comprehensive study of the measurement of students' abilities to solve problems covering the four years of an engineering program. Their study suggested that the ability to define a problem and to describe what he/she is doing as they solve the problem was a strong indication of the students' capacity to solve problems. Among their findings was:

- Students didn't realize that they should be learning about how to solve problems.
- A major deficiency of the students was their inability to define the problem.
- Students didn't understand that problem solving requires patience.

As a result they found that students would try to solve the problem before they fully understood the problem statement. The students in the ED&G 100 class showed the same deficiencies. Ultimately, using a series of questions to guide the students in the analysis of the problem statement was shown to be effective.

In class, students were first given the design problem statement [25] and asked to give a problem description and identify project goals, givens and unknowns. Students had limited previous experience with interpreting a problem statement through several in-class assignments. Thus, they did not have enough of the correct type of experience to properly complete this type of unstructured assignment. Many students gave answers that were often incomplete, especially for the project description. For instance, most students mentioned the stability and/or safety of the original cart but did not include specifics of why the original cart was unsafe or unstable.

A guided instructional approach was implemented that would require the students to provide more detailed answers. To encourage students to analyze the problem in detail and begin solution planning, the students were asked to answer the inquiry questions given in Fig. 2. Thus, question A was meant to help students understand the problem description. Their answers to this question, which were more detailed than the unstructured exercise previously described, included that the cart tips easily and is difficult to push. Consider the following portions of students' work when being asked to solve a problem (I) and when responding to the guided instructional approach (II).

Student 1 for Approach I wrote: 'The carts are not stable enough with the product being shipped around the plant. It is a hazardous work environment,' whereas for Approach II in response to Question A, the student wrote:

1. Tips over a lot.
2. Causes enormous danger to worker and factory.
3. Empty racks and partially filled racks occupy valuable floor space.
4. Partially filled racks occupy valuable space on the trucks as the aluminum parts are shipped for anodizing.

Student 2 for Approach I wrote: 'The current material handling system is unsafe and unstable,' whereas for Approach II in response to Question A, the student wrote:

1. Ergonomically incorrect—no handles.
2. Unstable—tipping.

Group Design Project
Write-up of individual ideas

A. List the problems (at least 4) with the current material handling system.

B. List the constraints for the design of the new material handling system.

C. List each of your ideas for the new material handling system. Explain how each of your ideas addresses each of the problems listed in part A.

D. Are each of your ideas consistent with the constraints listed in part B?

Fig. 2. Some inquiry questions used to guide students through implementation of the problem-solving heuristic to the design process.

3. Hard to load some shelves.
4. Wasted space because some shelves are incompletely/improperly loaded.

Questions C and D were intended as a beginning to the solution planning aspect of this project and as a tool to help students begin to relate the key points in the problem description to possible solutions. However, these inquiry questions need to be phrased properly to elicit specific answers. For question C, most students included some details relating back to the problem description in their answers, but many students were unable to give responses that were sufficiently complete to solve the problem without a great deal of difficulty. For question D half of the responses were simply 'yes'. Most students were unwilling, or not sufficiently motivated, to show how their ideas were consistent with the constraint. Question D may not have been directed enough to encourage the students to provide the required information. Even though the necessary calculations for their group design project were explicitly described, only about half of the design project groups performed these calculations on their proposed design, and none of the groups compared their calculations with those of the original cart.

DISCUSSION OF OUTCOMES

Since the background of students entering this course was quite diverse, no specific expectation of particular skills were assumed, including problem-solving skills. The problem-solving skills they developed previously would form the basis for new knowledge. Ideally, the students in the ED&G 100 course should be able to apply the problem-solving heuristic to almost any problem. However, they generally could not use the problem-solving heuristic unless a very specific example was first given in class. Then they followed the example very closely rather than modifying it to meet their own needs.

Among the major obstacles to improvement of students' problem-solving skills is the lack of understanding of the skills of students entering post-secondary education. Instructors assume students already have acquired the skills, and students believe that they are problem solvers when they either replicate the solution of one problem to solve another one, or 'plug' in all available data into an equation to get an answer to the problem [3]. The students in the ED&G 100 course completed a pre-course questionnaire in which 18 of the 20 students indicated that they were either good or excellent problem solvers. One indicated average and another one was not sure. Yet, in class they generally had difficulty in providing a clear statement of the problem.

Stating the problem clearly in a sentence or two often helps to identify the problem and may even present a possible solution. This statement, or question, identifies matters involving doubt, uncertainty, or difficulty as well as the limitations related to specific need or opportunity; it defines what the student wants or needs to know. Although the teacher can identify the problem, the student needs the experience of restating the problem in his or her own words as part of the skill development process. In the future, a problem-solving methodology could be more successfully implemented by beginning the semester with a guided decision-making approach while teaching students to ask useful inquiry questions. Future assignments throughout the semester can gradually transition to using an unstructured approach to successfully evaluate an engineering design problem. Students should be able to describe a problem clearly so someone else can understand it.

In addition to the difficulties students encountered in the problem statement component, other observations can be reported:

- Most students that did not have well thought-out solutions took longer to complete their assignments.
- Problems in the textbooks are generally not challenging enough for all students to find a problem-solving heuristic useful.

- All students, but especially the weaker ones, are very reluctant to write down anything while trying to solve the problem.

One of the most daunting shortcomings of most design textbooks is the lack of guided instructional problems that can be implemented in the classroom. Instructors often do not have the time to develop these types of assignments and creating a series of assignments designed to gradually develop problem-solving skills is likely to require some trial and error. In addition, since problem solving is perceived as a vague skill, assessing problem-solving skills can be perceived as too subjective and imperfect. Assessment methods [26] have been developed that seem to work well in assessing the ability of a student to apply a problem-solving heuristic on a quiz and have shown to provide an objective and consistent grading standard.

The positive impacts of use of the problem-solving heuristic were assessed on the final quiz of the course [2]. Students were instructed to develop a written solution design. A second section of ED&G 100, which was not exposed to this problem-solving heuristic, was given the same problem. All the students with the exposure to the problem-solving methodology were able to produce a step-by-step solution that directly related to implementation of their solution. On the other hand, many of the solutions given by the students in the second section were incomplete, and the steps did not have a one-to-one correspondence to problem requirements. In addition, more students in the first section clearly divided the main problem into subproblems and gave a solution for each subproblem. In addition, the students in the first section were more likely to annotate the picture in the problem statement and refer to their annotations in their solution method.

A significant outcome was that the students learned how to break down large problems into smaller subproblems. However, to get students to continue working on these skills, they would have to be reinforced in other classes. In reality, the regular use of this method by students in other classes was problematic depending on the nature of the courses they were enrolled in.

In order to improve the teaching of problem-solving skills in ED&G 100 in the future, a guided instructional approach with assignments gradually increasing in difficulty seems likely to be successful. Though problem-solving skills will be incorporated with the graphics component of the course, due to time constraints most of the assignments will relate to the design component. Initial exercises will consist of guided inquiry questions, which will lead students through the decomposition of the main problem into properly defined subproblems and the association of given data with the appropriate subproblem. In addition, the specific questions Woods *et al.* [5] suggest to help students identify and associate key points in the problem description with the problem solution should help students to interpret future problems on their own. Subsequent exercises would have students respond to more general questions and develop their own inquiry questions to help guide them through the solution process. Ideally, enough practice would be given throughout the semester such that students should be able to interpret the problem statement and apply the problem-solving heuristic on their own by the end of the semester. Also, requiring students to keep their notes and exercises in a problem-solving notebook would assist them when reviewing examples done in class and examining previous solutions to see if any give insight into a current problem.

To give students practice in properly defining engineering design problems, they should analyze several case studies throughout the semester. For the first case study, students would be guided through analysis of the problem with specific questions. On subsequent assignments, the students would be required to gradually fill in more of the specific details themselves. For instance, students could examine an article about the use of a strain gage to help determine how fast to deploy an airbag in a car crash [27]. The first assignment would include specific questions intended to lead the students to a clearly defined problem description. This assignment could include the following questions:

1. List three shortcomings of current airbags systems.
2. List three constraints the designers needed to consider when designing a new airbag system.
3. Explain how the engineers addressed each of the shortcomings listed in Question 1.

Then students can use their answers to these questions to create a problem description, which is specific enough to properly explain the problem.

CONCLUSION

The ABET Engineering Criteria 2000 is placing a greater emphasis on problem solving. It will require a very comprehensive effort on the part of engineering schools, including the assessment of the skills of incoming freshmen, professional development of faculty, and a continuous reinforcement of problem-solving skills throughout the curriculum in all technical courses. Though problem solving is often considered an important component in the engineering design process, it is often not given very much consideration in the classroom. Problem solving should be a key instructional technique throughout the course and the curriculum. Although each discipline has its own problem-solving approach, the common threads among them are strong. As students work to explore and solve the situations and problems presented to them in their courses, they can further develop and enhance strong critical thinking skills such as hypothesizing, planning, controlling variables, analyzing, interpreting, and assessing. Problem solving should become 'second nature' to students when they enter the work force after graduation. This paper shows how the methodology developed in an introductory computer science course [21] has been adapted to an introductory engineering design class.

A problem-solving heuristic, which has been implemented successfully in introductory computer science classes, was integrated into the solid modeling and design components of an introductory engineering class for one semester. Students were given experience dealing with problems that are not well defined. This study showed that many students have difficulty developing a good problem formulation in an unstructured assignment. Guided engineering instruction was used to help students deal with understanding these types of problems by taking them through the problem formulation and solution planning steps. In the future, initial problem solving will use guided engineering instruction and inquiry questions, so that by the end of the semester, students will be able to interpret and explain a problem description from an unstructured assignment.

Acknowledgements—The author would like to acknowledge the suggestions and advice of the other ED&G 100 faculty at Penn State Berks, Thomas H. Gavigan and Margaret Condon-Vance. This project was supported in part by the National Science Foundation, Award # CTS-0075076.

REFERENCES

1. A. Rugarcia, R. M. Felder, D. R. Woods, and J. E. Stice, The future of engineering education, I: A vision for a new century, *Chemical Engineering Education*, **34**(1), 2000, pp. 16–25.
2. S. J. Kimmel, F. P. Deek, and H. S. Kimmel, Applying problem-solving heuristics to a freshman engineering course, *Proc. 2001 American Society for Engineering Education Annual Conference*, Albuquerque, NM, June 24–27, 2001, in press.
3. Charles E. Wales, and A. H. Nardi, Do your students think or do they memorize? *Eng. Educ.*, **78**(7), 1988, pp. 682–688.
4. J. D. Burton, and D. M. White, Selecting a model for freshman engineering design, *J. Eng. Educ.*, **88**(3), 1999, pp. 327–332.
5. D. R. Woods, R. M. Felder, A. Rugarcia, and J. E. Stice, The future of engineering education, III: developing critical skills, *Chem. Eng. Educ.*, **34**(2), 2000, pp. 108–117.
6. G. W. Ernst, and A. Newell, *GPS: A Case Study in Generality and Problem Solving*, Academic Press, NY, 1969.
7. A. Newell, and H. A. Simon, *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ, 1972.
8. R. E. Mayer, *Thinking, Problem Solving, Cognition*, W. H. Freeman and Company, NY, 1983.
9. B. F. Skinner, An operant analysis of problem solving, in B.F. Skinner (ed.), *Problem Solving: Research, Method, and Theory*, John Wiley and Sons, NY, 1966.
10. H. A. Simon, *The Sciences of the Artificial*, MIT Press, Cambridge, MA (1969).
11. J. Piaget, *Genetic Epistemology*, Columbia University Press, NY (1970).
12. M. Rubinstein, *Patterns of Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ (1975).
13. H. A. Simon, Information-processing theory of human problem solving, in W. K. Estes (Ed.), *Handbook of Learning and Cognitive Processes*, Lawrence Erlbaum, Hillsdale, NJ (1978).
14. J. G. Greeno, Natures of problem-solving abilities, in W. K. Estes (ed.), *Handbook of Learning and Cognitive Processes*, Vol. 5, Lawrence Erlbaum, Hillsdale, NJ, 1978, pp. 239–270.
15. J. G. Greeno, A.M. Collins, and L. B. Resnick, Cognition and learning, in D. C. Berliner and R. C. Calfee (eds.), *Handbook of Educational Psychology*, Simon & Schuster Macmillan, NY (1996) pp. 15–45.
16. G. Polya, *How to Solve It*, Princeton University Press, Princeton, NJ (1945).
17. G. F. Wetzel, and W. G. Bulgren, *The Algorithmic Process: An Introduction to Problem Solving*, The SRA Computer Science Series, Chicago, IL (1985).
18. E. Dijkstra, *A Discipline of Programming*. Prentice-Hall, Englewood Cliffs, NJ (1976).
19. N. Wirth, Program development by stepwise refinement, *Communications of the ACM*, **14**(4), 1971, pp. 221–227.
20. F. P. Deek, *An Integrated Environment for Problem Solving and Program Development*, unpublished Ph.D. dissertation, New Jersey Institute of Technology. Newark NJ, 1997.
21. F. P. Deek, H. Kimmel and J. McHugh, Pedagogical changes in the delivery of the first course in computer science: problem solving then programming, *J. Eng. Educ.*, **87**(3), 1998, pp. 313–320.
22. F. P. Deek, M. Turoff and J. McHugh, A common model for problem solving and program development, *J. IEEE Trans. Educ.*, **42**(4), 1999, pp. 331–336.
23. J. A. Stuart, A method for teaching problem assessment, *Proc. 1997 Frontiers in Education Conf.*, Pittsburgh, PA, November 5–8, 1997, pp. 83–87.
24. D. R. Woods, C. M. Crowe, W T Hoffman, and J. D. Wright, Major challenges to teaching problem-solving skills, *Eng. Educ.*, **70**(3), 1979, pp. 277–284.
25. URL of problem statement: http://www.ecsel.psu.edu/~dhushy/EDG100/fa98_s1/design2.html. The actual problem statement given was modified to eliminate the requirement of a prototype and the mention of primary source information, due to constraints at the campus.
26. F. P. Deek, S. R Hiltz, H. Kimmel, and N. Rotter, Cognitive assessment of students' problem solving and program development skills, *J. Eng. Educ.*, **88**(3), 1999, pp. 317–326.
27. J. Degaspari, Out of Harm's Way, *Mechanical Engineering*, **122**(12), 2000, pp. 52–55.

Fadi P. Deek is Associate Dean of the College of Computing Sciences, Director of the Information Technology Program, and Associate Professor of Information Systems at NJIT. He is also a member of the Graduate Faculty Ph.D. Program in Management, Rutgers University. Dr. Deek's research interests include learning systems, software engineering, programming environments, problem solving/cognition/learning theory, and computer science education. Dr. Deek has published over 90 papers in journals and conference proceedings and he has given over 40 invited and professional presentations. He is also the author of the book Computer-Supported Collaboration with Applications to Software Development (Kluwer Academic Publishers, 2003).

Shari Kimmel has been teaching Engineering Design and Graphics 100 at Penn State Berks-Lehigh Valley College. She received a BS in mechanical engineering from Rutgers University and a Ph.D. in aerospace engineering from the University of Southern California. She worked as a post-doc at Penn State University—University Park. Dr. Kimmel was a visiting research scientist and the PI for a NSF grant at Lehigh University and Assistant Professor of Mechanical Engineering at Temple University. Currently, she is a visiting research scientist at the University of Cincinnati.

Howard Kimmel, Professor of Chemical Engineering and Assistant Vice President for Academic Affairs at New Jersey Institute of Technology, has spent the past twenty-seven years designing and implementing professional development programs and curricula for K-12 teachers in science and technology. At the college level, he collaborates on projects exploring teaching methodologies and assessment strategies in first year college courses in the sciences, engineering, and computer science.