1

What Is Problem Solving?

LEARNING IN THE everyday world, where people live and work, is omnipresent and essential to survival, let alone progress. In homes, businesses, organizations, and societies in every culture, learning is driven by problems that need solving. How do I pay for a new car? Which schools should my children attend? How do we design a new marketing campaign to address a target market? How do we make peace with our enemies? What's wrong with the compressor? How do we raise funds to support municipal services?

Modern life in nearly every context presents a deluge of problems that demand solutions. Although many trainers avoid using the word *problem* because it implies acquiescence and insolubility (a problem with problem solving is that *problem* has many meanings), intellectually that is what they get paid to do. Designing training is an archetype of design problem solving. And most of these problems that people face in their everyday lives are ill structured. They are not the well-structured problems that students at every level of schooling, from kindergarten through graduate school, attempt to solve at the back of every textbook chapter.

The ubiquity of problems in our lives and the limited amount of time that always seems to be allocated to education and learning lead me to argue two things. First, telling students what we know about the world and quizzing their recall of what we told them is not only an insult to our learners (we should expect more of them); that pedagogy also retards their epistemological development, preventing them from developing the knowledge-seeking skills they need (Jonassen, Marra, and Palmer, 2004).

The second point that I argue is that the only legitimate goal of education and training should be problem solving. Why? Because people need to learn how to solve problems in order to function in their everyday and professional lives. No one in the everyday world gets paid for memorizing facts and taking exams. Most people get paid for solving problems. Content, the coin of the educational realm, is relatively meaningless outside the context of a problem. From kindergarten through graduate school, students study content without clear purpose or reason. If they studied content for the explicit purpose of solving problems, the content would have more meaning. Second, what is learned in the context of solving problems is better comprehended and better retained. Some educators, however, believe that if education is focused on solving problems, students will miss the breadth of learning that is reflected in the curriculum. That is probably true, but they will learn more.

Let us compare a couple of learning equations. Students who memorize information for the test usually retain less than 10 percent of what they learn, so 10 percent of the whole curriculum (100 percent assuming that the teacher or trainer can cover the whole curriculum) yields a 10 percent learning outcome (and it is probably less than that). In a problem-oriented curriculum, students may cover only 50 percent of the curriculum, but they understand and remember 50 percent of what they learn, yielding a 25 percent learning outcome. These figures cannot be validated in different contexts, but the point is simple: when students are solving problems, they learn and comprehend more. Remember the most important lessons that you have learned in your life. They probably resulted from solving some kind of problem.

What Are Problems, and How Do They Vary?

Just what is a problem? There are at least two critical attributes in my definition of a problem. First, a problem is an unknown entity in some context (the difference between a goal state and a current state). Second, finding or solving for the unknown must have some social, cultural, or intellectual value. That is, someone believes that it is worth finding the unknown. If no one perceives an unknown or a need to determine an unknown, there is no perceived problem.

There are a number of variable attributes of problems. Problems vary in knowledge needed to solve them, the form they appear in, and the processes needed to solve them. The problems themselves also vary considerably, from simple addition problems in elementary school to complex social-cultural-political problems like those encountered in the Middle East. Intellectually, problems vary in at least four ways: structuredness, complexity, dynamicity, and domain specificity or abstractness.

Structuredness

Problems within domains and between domains vary in terms of how well structured they are. Jonassen (1997) described problems on a continuum from well structured to ill structured. The most common problems that students solve in schools, universities, and training venues are well-structured problems. Like the story problems found at the end of textbook chapters or on examinations, well-structured problems require the application of a limited and known number of concepts, rules, and principles being studied within a restricted domain. They have a well-defined initial state, a known goal state or solution, and a constrained set of logical operators (a known procedure for solving). Well-structured problems also present all elements of the problem to the learners, and they have knowable, comprehensible solutions.

Ill-structured problems, at the other end of the continuum, are the kinds of problems that are more often encountered in everyday and professional practice. Also known as wicked problems, these problems do not necessarily conform to the content domains being studied, so their solutions are neither predictable nor convergent. Ill-structured problems are also interdisciplinary, that is, they cannot be solved by applying concepts and principles from a single domain. For example, solutions to problems such as local pollution may require the application of concepts and principles from math, science, political science, sociology, economics, and psychology. Ill-structured problems often possess aspects that are unknown (Wood, 1983), and they possess multiple solutions or solution methods or often no solutions at all (Kitchner, 1983). Frequently, multiple criteria are required for evaluating solutions to ill-structured problems, and sometimes the criteria are not known at all. Ill-structured problems often require learners to make judgments and express personal opinions or beliefs about the problem.

For a long time, psychologists believed that "in general, the processes used to solve ill-structured problems are the same as those used to solve well structured problems" (Simon, 1978, p. 287). However, more recent research in everyday problem solving in different contexts makes clear distinctions between thinking required to solve well-structured problems and everyday problems. Dunkle, Schraw, and Bendixen (1995) concluded that performance in solving well-defined problems is independent of performance on ill-defined tasks, with ill-defined problems engaging a different set of epistemic beliefs. Hong, Jonassen, and McGee (2003) showed that solving ill-structured problems in a simulation called on different skills than well-structured problems did, including the use of metacognition and argumentation (see Chapter Six). Other studies have shown differences in required processing for well-structured and ill-structured problems. For example, communication patterns among problem solvers differed while teams solved well-structured versus ill-structured problems (Jonassen and Kwon, 2001). Groups that solved ill-structured problems produced more extensive arguments in support of their solutions when solving ill-structured problems because of the importance of generating and supporting alternative solutions (Cho and Jonassen, 2002).

Although the need for more research comparing well-structured and ill-structured problems is obvious, it seems reasonable to predict that different intellectual skills are required to solve well-structured than ill-structured problems, and therefore the ways that we teach people to solve well-structured problems cannot be used effectively to teach people to solve ill-structured problems. Probably some very ill-structured problems cannot be taught at all. They must be experienced and dealt with using general intelligence and world knowledge.

Complexity

Problems vary in terms of their complexity. Problem complexity is determined by the number of issues, functions, or variables involved in the problem; the degree of connectivity among those variables; the type of functional relationships among those properties; and the stability among the properties of the problem over time (Funke, 1991). Simple problems, like textbook problems, are composed of few variables, while ill-structured problems may include many factors or variables that may interact in unpredictable ways. For example, international political problems are complex and unpredictable. Complexity is also concerned with how many, how clearly, and how reliably components are represented in the problem. We know that problem difficulty is related to problem complexity (English, 1998). The idea of problem complexity seems to be intuitively recognizable by even untrained learners (Suedfield, de Vries, Bluck, and Wallbaum, 1996). The primary reason is that complex problems involve more cognitive operations than simpler ones do (Kluwe, 1995). Balancing multiple variables during problem structuring and solution generation places a heavy cognitive burden on problem solvers.

Complexity and structuredness overlap. Ill-structured problems tend to be more complex, especially those emerging from everyday practice. Most well-structured problems tend to be less complex; however, some well-structured problems can be extremely complex and ill-structured problems can be fairly simple. For example, video games can be very complex well-structured problems, while selecting what to wear from our closet for different occasions is a simple ill-structured problem (at least for some of us).

Dynamicity

Problems vary in their stability or dynamicity. More complex problems tend to be dynamic; that is, the task environment and its factors change over time. When the conditions of a problem change, the solver must continuously adapt his or her understanding of the problem while searching for new solutions, because the old solutions may no longer be viable. For example, investing in the stock market is often difficult because market conditions (for example, demand, interest rates, or confidence) tend to change, often dramatically, over short periods of time. Static problems are those where the factors are stable over time. Ill-structured problems tend to be more dynamic, and well-structured problems tend to be fairly stable.

Domain (Context) Specificity/Abstractness

Most contemporary research and theory in problem solving claims that problem-solving skills are domain and context specific. That is, problem-solving activities are situated, embedded, and therefore dependent on the nature of the context or domain knowledge. Mathematicians solve problems differently from engineers, who solve problems differently from political scientists, and so on. Problems in one organizational context are solved differently than they are in another context. Problems at IBM are solved differently from those at Hewlett-Packard. They have different organizational structures, different cultures, and different sociological mixes, all of which affect the kinds of problems that arise and how they are solved. Problems within a domain rely on cognitive operations that are specific to that domain (Mayer, 1992; Smith, 1991; Sternberg and Frensch, 1991). For example, students in the probabilistic sciences of psychology and medicine perform better on statistical, methodological, and conditional reasoning problems than do students in law and chemistry, who do not learn such forms of reasoning (Lehman, Lempert, and Nisbett, 1988). The cognitive operations required to solve problems within a domain or context are learned through the development of pragmatic reasoning rather than results from solving that kind of problem. Individuals in different domains or contexts develop reasoning skills through solving ill-structured problems that are situated in those different domains or contexts and require forms of logic that are specific to that domain or context.

In sum, problems within a domain or context vary in terms of their structuredness, complexity, and dynamicity, but all problems vary also along another dimension between domains or contexts. Which affects problems more, context or problem type, is not known.

What Is Problem Solving, and How Does It Vary?

If a problem is an unknown worth solving, then problem solving is "any goal-directed sequence of cognitive operations" (Anderson, 1980, p. 257) directed at finding that unknown. Those operations have two critical attributes. First, problem solving requires the mental representation of the problem and its context. That is, human problem solvers construct a mental representation (or mental model) of the problem, known as the *problem space* (Newell and Simon, 1972). Although there is little agreement on the meaning of mental models or problem spaces, internal mental models (as opposed to social or team mental models) of problems are multimodal representations consisting of structural knowledge, procedural knowledge, reflective knowledge, images and metaphors of the system, and executive or strategic knowledge (Jonassen and Henning, 1999). That is, mental models consist of knowledge about the structure of the problem, knowledge of how to perform tests and other problem-solving activities, the envisionment of the problem and its constituent parts (De Kleer and Brown, 1981), and knowledge of when and how to use procedures. The mental models of experienced problem solvers integrate these different kinds of knowledge, and it is the mental construction of the problem space that is the most critical for problem solving. Second, successful problem solving requires that learners actively manipulate and test their models. Thinking is internalized activity (Jonassen, 2002), especially when solving problems, so knowledge and activity are reciprocal, interdependent processes (Fishbein and others, 1990). We know what we do, and we do what we know. Successful problem solving requires that learners generate and try out solutions in their minds (mental models or problem spaces) before trying them out in the physical world.

If problems differ in terms of structure, complexity, and context, then so too must the kinds of problem-solving processes. How many kinds of problem solving are there? Jonassen (2000a) described a typology of problems, including puzzles, algorithms, story problems, rule-using problems, decision making, troubleshooting, diagnosis-solution problems, strategic performance, systems analysis, design problems, and dilemmas. Table 1–1 describes characteristics of each kind of problem solving. Note that this typology (not taxonomy) described my mental model of problem solving in the year 2000. Additional research and experience may verify more or fewer kinds of problems.

Table 1–1. Kinds of Problems.

Problem Type	*Logical Problem*	*Algorithm*	*Story Problem*	*Rule-Using Problem*	*Decision Making*
Learning activity	Logical control and manipulation of limited variables; solve puzzle	Procedural sequence of manipulations; algorithmic process applied to similar sets of variables; calculating or producing correct answer	Disambiguate variables; select and apply algorithm to produce correct answer using prescribed method	Procedural process constrained by rules; select and apply rules to produce system-constrained answers or products	Identifying benefits and limitations; weighting options; selecting alternative and justifying
Inputs	Puzzle	Formula or procedure	Story with formula or procedure embedded	Situation in constrained system; finite rules	Decision situation with limited alternative outcomes
Success criteria	Efficient manipulation; number of moves or manipulations required	Answer or product matches in values and form	Answer or product matches in values and form; correct algorithm used	Productivity (number of relevant or useful answers or products)	Answer or product matches in values and form
Context	Abstract task	Abstract, formulaic	Constrained to predefined elements, shallow context	Purposeful academic, real world, constrained	Life decisions
Structuredness	Discovered	Procedural predictable	Well-defined problem classes; procedural predictable	Unpredicted outcome	Finite outcomes
Abstractness	Abstract, discovery	Abstract, procedural	Limited simulation	Need based	Personally situated

Trouble-shooting	*Diagnosis-Solution*	*Strategic Performance*	*Case Analysis*	*Designs*	*Dilemmas*
Examine system; run tests; evaluate results; hypothesize and confirm fault states using strategies (replace, serial elimination, space split)	Troubleshoot system faults; select and evaluate treatment options and monitor; apply problem schemas	Applying tactics to meet strategy in real time; complex performance maintaining situational awareness	Solution identification, alternative actions, argue position	Acting on goals to produce artifact; problem structuring and articulation	Reconciling complex, nonpredictive, vexing decision with no solution; perspectives irreconcilable
Malfunctioning system with one or more faults	Complex system with faults and numerous optional solutions	Real-time, complex performance with competing needs	Complex, leisure-time system with multiple ill-defined goals	Vague goal statement with few constraints; requires structuring	Situation with antinomous positions
Fault(s) identification; efficiency of fault isolation	Strategy used; effectiveness and efficiency of treatment; justification of treatment selected	Achieving strategic objective	Multiple, unclear	Multiple, undefined criteria; no right or wrong, only better or worse	Articulated preference with some justification
Closed system, real world	Real world, technical, mostly closed system	Real-time performance	Real world, constrained	Complex, real-world degrees of freedom; limited input and feedback	Topical, complex, interdisciplinary
Finite faults and outcomes	Finite faults and outcomes	Ill-structured strategies; well-structured tactics	Ill-structured	Ill-structured	Finite outcomes, multiple reasoning
Problem situated	Problem situated	Contextually situated	Case situated	Problem situated	Issue situated

Source: From Jonassen (2000a).

Regardless of how many kinds of problems there are, I believe that there are similarities in the cognitive processing engaged within these classes of problems. Within classes, there are differences in problem solving depending on the domain or context in which the problem occurs and its structuredness and complexity. Because it is practically impossible to design and develop models, methods, and tools for solving problems in every domain, this book focuses on three different kinds of problems. My goal is to show how methods for representing problems, assessing solutions, and designing learning environments vary across problem types because one of the underlying principles of instructional design is that different learning outcomes engage different learning processes and therefore require different conditions of learning (Gagné, 1960). I want to show how these problem types differ and how instruction to support them should also differ. However, space limitations prevent illustrating models and methods for each of the eleven types of problems identified in the typology in Table 1-1, and, frankly, I have not constructed all of those models yet. So I have chosen three commonly encountered yet divergent kinds of problems to illustrate methods throughout the book: story problems, troubleshooting problems, and case or system or policy analysis problems. I next describe each of these kinds of problems and in Chapter Two describe models for teaching students how to solve each of these three kinds of problems. Chapters Three through Six elucidate the methods described in Chapter Two. Chapter Seven describes a variety of methods to help students reflect on problem-solving processes because reflection is essential to meaningful learning. Finally, Chapter Eight then describes how to assess problem solving for each kind of problem.

Story Problems

From simple problems in beginning mathematics to complex story problems in engineering dynamics classes, story problems are the most commonly used and extensively researched kind of problems. Found at the back of thousands of textbook chapters, these problems are usually, though not most effectively, solved by learners by identifying key words in the story, selecting the appropriate algorithm and sequence for solving the problem, applying the algorithm, and checking their responses, which they hope will be correct (Sherrill, 1983). Story problem solving requires not only calculation accuracy but also the comprehension of textual information, the capacity to visualize the data, the capacity to recognize the semantic structure of the problem, the capacity to sequence their solution activities correctly, and the capacity and willingness to evaluate the procedure that they used to solve the problem (Lucangelli, Tressoldi, and Cendron, 1998).

What do these mean? Different kinds of story problems have different semantic structures, so successfully solving these problems requires that learners develop semantic models of the deep structure of the problem as well as a model of the processing operations required to solve the problem (Riley and Greeno, 1988). Solving story problems requires significant conceptual understanding of the problem class. Based on an extensive analysis of story problem solving, Jonassen (2003) found that solving any kind of story problem requires that learners construct a mental model of the problem type that includes a model of the situation depicted in the surface content, as well as a semantic model of the structure of the problem. For example, simple mathematics motion problems typically use trains, cars, or airplanes traveling in one direction or another as the surface content. In order to be able to solve motion problems, the learner relates trains, cars, and planes to a semantic model of the relationships between the different entities in a problem. For example, there are different kinds of motion problems, such as overtake (one vehicle starts and is followed later by a second, which travels over same route at faster rate), opposite direction (two vehicles leaving the same point are traveling in opposite directions), round trip (a vehicle travels from point A to B and returns), or closure (two vehicles start at different points traveling toward one another) (Mayer, Larkin, and Kadane, 1984). Each kind of motion problem has a different set of structural relations between the problem entities that call on different processing operations. Story problems require that learners understand the situational and structural relationships between the problem entities. Associating situational and structural models leads to comprehension of different classes of story problems. These classes of problems are domain specific. Chemistry problems have different situations and structures than physics problems do, which differ from biology problems.

An analysis of the cognitive requirements for solving story problems shows that learners must do the following things:

- Parse the problem statement, that is, read and break down the description of the problem.
- Try to classify the problem type by:

  Comparing the surface content of the problem to problems previously solved or to problem class descriptions.

  Comparing structural relationships described in the problem to problem models or to previously solved problems.
- Construct a mental representation of the problem being solved by:

  Identifying problem entities (sets) from the surface content.

  Mapping those sets onto the structural model of the problem.

  Accessing the formula and processing operations required to solve the problem.
- Map the values in each set onto the formula.
- Estimate the size of the solution and the proper units (distance, length, and so forth).
- Solve the formula.
- Reconcile the value with the estimate in terms of size and units. (Was the result similar to the estimate?)
- Remember the problem content and the structure of the problem entities and file according to problem type.

In Chapter Two, I will describe a model for designing instruction to support learning to solve story problems.

Troubleshooting Problems

Troubleshooting is among the most commonly experienced kinds of problem solving in the professional world. From troubleshooting a faulty modem to a multiplexed refrigeration system in a modern supermarket, troubleshooting attempts to isolate fault states in some dysfunctional system. Once the fault is found, the part is replaced or repaired.

Troubleshooting is often thought of as a linear series of decisions that leads to fault isolation. That approach may work for helping novices solve simple troubleshooting problems, but it is inadequate for training competent, professional troubleshooters because troubleshooting is not merely a series of decisions. Effective troubleshooting requires system knowledge (conceptual knowledge of how the system works), procedural knowledge (how to perform problem-solving procedures and test activities), and strategic knowledge (knowing when, where, and why to apply procedures) (Pokorny, Hall, Gallaway, and Dibble, 1996). These components comprise the troubleshooter's mental model of the process, which consists of conceptual, functional, and declarative knowledge, including knowledge of system components and interactions, flow control, fault states (fault characteristics, symptoms, contextual information, and probabilities of occurrence), and fault testing procedures. These skills are integrated and organized by the troubleshooter's experiences.

The best predictor of a troubleshooter's skills is the number of similar problems that she or he has solved. Learning to troubleshoot is best facilitated by experience. Technicians through physicians can recall with extraordinary accuracy problems that they have troubleshot many years before. The problems that are most completely and accurately recalled are those that are most difficult to solve, because the problem solver was more conceptually engaged in the process. The primary differences between expert and novice troubleshooters are the amount and organization of system knowledge (Johnson, 1988). An analysis of the cognitive processes required to solve troubleshooting problems shows that learners must:

- Identify the fault state and related symptoms, that is, define the current state of the system being troubleshot.
- Construct a mental model of the problem by:

    Describing the goal state (how do you know when system is functioning properly).

    Identifying the faulty subsystem (known as space splitting).

- Diagnose the problem by:

  Examining the faulty subsystems;

  Remembering previously solved problems;

  Reusing or adapting the previously solved problem;

  If no previously solved problem is available, ruling out the least likely hypotheses;

  Generating an initial hypothesis and assumptions about the problem;

  Testing this hypothesis based on domain knowledge;

  Interpreting the results of the test;

  Confirming or rejecting the validity of the hypothesis, and if it is rejected, generating a new hypothesis;

  Repeating the process of generating and testing hypotheses until the fault is identified.

- Implement the solution by replacing the defective part or subsystem.

  Test the solution to determine if the goal state is achieved.

- Record the results in a fault database (that is, remember the case for future reuse).

Case and System and Policy Analysis Problems

Case, system, or policy analysis problems (hereafter referred to as case problems) tend to be complex and ill-structured policy or analysis problems. Case analysis problems emerged at Harvard Law School nearly 130 years ago (Williams, 1992). Analyzing legal cases, preparing briefs, and defending judgments are authentic case analysis problems for law students. In business and many other professional contexts, such as international relations (Voss, Wolfe, Lawrence, and Engle, 1991) and managerial problem solving (Wagner, 1991), analyzing complex, situated case problems defines the nature of work. Business problems, including production planning, are common case problems. Classical situated case problems also exist in international relations, such as, "Given low crop productivity in the Soviet Union, how would the solver go about improving crop productivity if he or she served as Director of the Ministry of Agriculture in the Soviet Union?" (Voss and Post, 1988, p. 273). International relations problems involve situational analysis, decision making, solution generation, and testing in complex and dynamic political contexts.

Case and systems analysis problems are usually found everywhere except in the classroom, usually because they are complex and ill structured and therefore not amenable to easy assessment. Pick up any newspaper, and within the first few pages, you will find numerous case analysis problems:

- Where to locate a new municipal landfill
- How to develop a policy for rent control in Chicago
- How to pass a new funding law in a parliamentary country
- What to advise the president on political strategies in the Middle East
- How to resolve or at least mitigate racial prejudices in Malaysian schools
- How to encourage biodiversity in Third World countries
- What levels of farm subsidies to recommend in the Midwest

In addition to finding problems in newspapers or news magazines, you may wish to examine the Web site of the Union of International Associations (www.uia.org), an organization that maintains a database of thirty thousand problems around the world.

Case, system, or policy analysis problems are usually complex and interdisciplinary. That is, a reasonable solution is impossible by examining the problem from a single viewpoint. The problems set out in the list above all require economic, political, sociological, psychological, anthropological, and various scientific (engineering, chemical, biological) perspectives for their solution. Too often, case problems are insoluble because we focus too narrowly on situating the problem within a single domain. Is it a political, sociological, or economic problem? Likely it is all three.

In case analysis problems, goals are vaguely defined in the problem statement. Often, a significant part of the problem is understanding what the real problem is. No constraints may be stated, and little is known about how to solve the problem. There is usually no consensual agreement on what constitutes a good solution. The information available to the problem solver may be prodigious but may also be incomplete, inaccurate, or ambiguous (Voss, Wolfe, Lawrence, and Engle, 1991). Case analysis problems are very ill structured. Therefore, "the whole process of coping with a complex problem can be seen as a process of intention regulation" (Dörner and Wearing, 1995), that is, deciding what needs to be done. To complicate case analysis problems, "there are no formal procedures or guidelines to govern case analysis or evaluation of problem solutions," and what skilled performers need to know in order to solve these complex case problems is often tacit (Wagner, 1991, p. 179).

Solving case analysis problems cannot be as clearly defined as story problem solving or troubleshooting because the problems are ill structured. Bardach (2000) claims that solving policy problems is an eightfold process: define the problem, assemble some evidence, construct the alternatives, select the criteria, project the outcomes, confront the trade-offs, decide, and tell your story. Some of these activities are used for designing case analysis instruction. It is difficult to enumerate a process for case analysis problems; however, it generally starts with problem identification, followed by contextual analysis that involves a lot of information collection. When analyzing a complex situation, the problem solver intentionally seeks to identify the multiple personal, disciplinary, and thematic perspectives that may define the case. Who are the stakeholders, and what beliefs and perspectives do they bring to the case? The problem solver must also reconcile those perspectives into a solution; forecast outcomes (predicting effects); plan for implementation, which involves a lot of decision making; monitor the effects of one's actions; and reflect on the efficacy of the solution (Dörner and Wearing, 1995).

The final steps, implementation and assessment, are often impossible to accomplish in case problems. For example, it is not possible to try out a new political strategy on the Middle East and see what responses occur. Even if we could, the results could prove disastrous. Therefore, case analysis problems in education settings usually end with an argumentation stage. That is, the problem solvers will generate a solution and then argue for that solution (see Chapter Six). This process of justification provides valuable assessment information (see Chapter Eight). Sometimes it is the only form of assessment of problem-solving skills. Case analysis problems are among the most contextually dependent kind of problem solving, so analyzing cases places a much greater importance on situation analysis.

Why solve case analysis problems in schools or training situations, especially if they cannot be solved? Solving these problems is a lot more conceptually engaging than memorization. Solving case analysis problems engages learners in understanding and resolving the issues rather than remembering them. Resolving case analysis problems requires that learners critically analyze situations, identify issues and assumptions underlying different positions, consider consequences, use cognitive flexibility, and engage in reflective, ethical decision making (Lundeberg, 1999). These are all goals that educators espouse but too seldom engage in. Getting students to learn how to deal with ambiguity alone is a valuable goal in itself. Any teacher or professor who tires of students' asking what will be on the next test needs to engage his or her students in solving case, systems, or policy analysis problems. The levels of learning and thinking engaged by the process are at a much deeper level and are more meaningful. Although a quantifiable problem solution may not be possible, understanding the world or context that we live and function in helps us to construct much richer mental models of that world. If you are training business managers for international operations, for example, understanding the problems of the cultures in which they will operate will improve their abilities to manage. Many educational mission statements cite the importance of students' becoming better informed and more engaged citizens. Solving case analysis problems will help students to achieve that mission.

SUMMARY

Almost all learning in everyday and professional contexts (what some people refer to as the real world) is driven by the need to solve a problem, whether you admit it or not. Those who are better able to learn in order to solve problems have been more successful throughout history. Cave dwellers had to solve problems in order to survive environmental threats. Egyptians solved some very complex problems in order to build the pyramids. Wars throughout history can be viewed as problem-solving contests. In modern business, the best problem solvers dominate markets. Engineers, builders, marketers, chemists, politicians, social workers, and maintenance workers are paid to solve problems. In contemporary homes, the best problem solvers lead the most fulfilled lives. Problem solving is a major part of our everyday experience, and it is found everywhere *except* in schools, universities, and training organizations. When I have made this point to teachers, university faculty, and corporate trainers around the world, their reactions have varied from uneasiness to hostility. Perhaps they perceive problems as mysterious, confrontational, or impossible. Perhaps they do not know how to solve problems themselves. It is time that they learned.

If solving problems is the predominant intellectual skill required by workers and people in nearly every setting in the world (several corporate and institutional reports have made that claim), instructional designers should be developing models and methods for helping learners to become more effective problem solvers. When we lecture to students or trainees, we may feel good about what we have taught, but you can bet that the learners have learned (that is, made meaning of) little, if anything. Requiring students to memorize information for a test insults the learners and prevents them from becoming intellectually capable thinkers. When we learn something in the context of solving a problem, we understand and remember it better. If instructional designers do not begin to include problem solving as part of their instruction to students and trainees, then they are wasting their own time and students' time. In order to engage students in problem solving, we do not have to give them the basics before they can solve problems, for two reasons: first, we cannot give knowledge, and second, it is wishful thinking to hope that learners can take the basics and learn to solve problems when we have neither taught them nor even given them the opportunity. As educators and trainers, we have an obligation. We need to get started.

This book seeks to disseminate what I have learned about problem solving. It is only a beginning. There is so much more to learn. I hope that it helps you to confront some of your instructional problems.