Instructional Design Models for Well-Structured and Ill-Structured Problem-Solving Learning Outcomes

☐ David H. Jonassen

Although problem solving is regarded by most educators as among the most important learning outcomes, few instructional design prescriptions are available for designing problem-solving instruction and engaging learners. This paper distinguishes between well-structured problems and ill-structured problems. Well-structured problems are constrained problems with convergent solutions that engage the application of a limited number of rules and principles within well-defined parameters. Ill-structured problems possess multiple solutions, solution paths, fewer parameters which are less manipulable, and contain uncertainty about which concepts, rules, and principles are necessary for the solution or how they are organized and which solution is best. For both types of problems, this paper presents models for how learners solve them and models for designing instruction to support problem-solving skill development. The model for solving well-structured problems is based on information processing theories of learning, while the model for solving ill-structured problems relies on an emerging theory of ill-structured problem solving and on constructivist and situated cognition approaches to learning.

PROBLEM: INSTRUCTIONAL-DESIGN MODELS FOR PROBLEM SOLVING

☐ Most educators agree that problem solving is among the most meaningful and important kinds of learning and thinking. However, most taxonomies of learning and instructional design models do not even acknowledge it as a learning outcome. After abandoning problem solving by name in his earlier taxonomy (Gagné, 1977), Gagné (1985) later regarded problem solving as the synthesis of other rules and concepts into higher-order rules, which can be applied in a constrained set of situations. Since Bloom's taxonomy (Bloom, Engleshart, Furst, Hill, & Krathwohl, 1956) was developed to classify test outcomes and not learning outcomes, we must infer that problem solving would require a combination of analysis and synthesis skills, though it is not specifically identified. Component display theory (Merrill, 1983) does not acknowledge problem solving, so we must infer that it includes "using" and "finding" rules and principles. Traditional, hierarchical models of learning and instructional design have assumed that problem solving is composed of building blocks, such as concepts, rules, and principles, that are called on by learners when faced with a problem.

Problem solving, as an activity, is more complex than the sum of its component parts. Without question, problem solving necessarily engages a variety of cognitive components, such as propositional information, concepts, rules, and principles (*domain knowledge*). However, it also involves *structural knowledge* (information networking, semantic mapping/

conceptual networking, and mental models), *ampliative skills* (constructing/applying arguments, analogizing, and inferencing), and *metacognitive skills* (goal setting, allocating cognitive resources, assessing prior knowledge, assessing progress/error checking) in the learner. Additionally, problem solving also engages *motivation/attitudinal components* (exerting effort, persisting on task, engaging intentionally) and certainly requires *knowledge about self* (articulating prior knowledge, articulating sociocultural knowledge, articulating personal strategies, and articulating cognitive prejudices/weaknesses) (Jonassen & Tessmer, 1996).

Because problem solving outcomes are not sufficiently acknowledged or articulated in the instructional-design literature, little advice about how to design problem-solving instruction is available. Generic recommendations about using case instruction, simulations, Socratic dialogues, heuristics, and algorithms to engage and support problem solving are common, but no instructional-design models provide any prescriptions for designing the components of instruction. Gagné, Briggs, and Wager (1992) acknowledge that problem-solving learning is difficult and suggest only a brief template for applying the *events of instruction* in the same way they treat concept learning and rule learning outcomes. Smith and Ragan (1993) also suggest the events of instruction model, although they acknowledge differences in the nature of the steps. None of the other instructional design texts even acknowledges problem solving, let alone prescribes methods for supporting it. Yet the most pervasive assumption of instructional design is that different learning outcomes necessitate different conditions of learning (Gagné, 1966). So, instruction to support problem-solving learning outcomes should differ from that used to support, for instance, concept learning or rule learning. Assuming that and assuming that problem solving is more than the sum or even the synthesis of its component skills, specific models of problem-solving instruction need to be proposed and tested.

This article proposes instructional design models for supporting learning how to solve a range of problems, from well-structured problems to ill-structured problems. Instructional designs for well-structured problems are rooted in information processing theory while instructional designs for ill-structured problems necessarily borrow assumptions and methods from constructivism and situated cognition. Information processing theories conceive of learning outcomes as generalizable skills that can be applied to any content domain, while constructivism and situated cognition claim that problem solving is domain- and context-dependent (Bransford, 1994) and therefore constrained by context. So, problem solving in different contexts and domains calls upon different skills.

ATTRIBUTES OF PROBLEMS

In the most general sense, a problem is an unknown that results from any situation in which a person seeks to fulfill a need or accomplish a goal. However, problems are problems only when there is a "felt need" that motivates people to search for a solution in order to eliminate discrepancies (Arlin, 1989).

Problems are traditionally defined by a problem *domain*, a problem *type*, a problem-solving *process*, and a *solution*.

The problem domain consists of the content (concepts, rules, and principles) that defines the problem elements.

The problem type describes the combination of concepts and rules and the procedures for acting on them in order to solve the problems. For instance, oxidation reactions in chemistry are a type of problem that are solved in a similar manner.

The problem-solving process depends upon the problem solver's understanding and representation of the type problem, including an understanding of the problem state and goal state. These, along with a set of operators for moving from the initial state to the goal state, are known as the problem *space* or problem *schema* (Wood, 1983). The problem space is "the fundamental organizational unit of all human goal-oriented activity" (Newell, 1980, p. 696). With practice over time, problem solvers construct richer problem representations or schemas which they can apply in a more proceduralized or automatized manner. Therefore, experts differ from novices because their problem schemas better enable them to recognize a problem situation as belonging to a certain class of problem. Novices, on the other hand, possess deficient problem schemas and so are not able to recognize problem states as well, so they have to rely on generalized problem-solving strategies (Sweller, 1988).

The solution to the problem represents the goal of the problem solver. The solution may be convergent (a single, known solution), or it may be divergent (one of several acceptable solutions). A critical attribute of problem solving is that the solution to the problem is not readily apparent or specified in the problem statement, so the learner must identify not only the nature of the problem, but also an acceptable solution, and a process for arriving at it.

KINDS OF PROBLEMS

The kinds of problems that humans solve vary dramatically, as do the nature of problem situations, solutions, and processes. The domain, goal, and processes entailed by a problem may be very well-structured (e.g., solving a quadratic equation, identifying molar equivalents, providing balanced lighting for a stage set) or it may be very ill-structured (e.g., designing an addition to your home, figuring out whether or not to trade in your ten-year-old car, losing weight, or prescribing an instructional solution for a given problem situation). Problems may also be presented or discovered, well-defined or ill-defined, simple or complex, long-term or short term, and familiar or unfamiliar (Arlin, 1989). These varieties of problems cluster into three kinds of problems: *puzzle* problems, *well-structured* problems, and *ill-structured* problems. These problem types do not represent well-defined classifications, but rather represent a continuum from decontextualized problems with convergent solutions to very contextualized problems with multiple solutions.

Puzzle Problems

Most problem-solving research has examined solutions to a class of decontextualized problems that are designed to manifest reasoning and thinking processes. These include content-neutral puzzles, such as anagrams, the Tower of Hanoi problem (Simon, 1976), the Nine Dots problems (Chi & Glaser, 1985) or the Missionaries and Cannibals problem, in which you start with five missionaries and five cannibals and a boat that can hold three people. The goal is to transport them all across the river in groups of one to three in the boat, with the caveat that cannibals should never outnumber missionaries in any place. The optimal solution requires eleven moves, though most people usually require twenty or more. Puzzle problems are well-structured with a single correct answer where all elements required for the solution are known and solutions require using logical, algorithmic processes (Kitchner, 1983) such as means-ends analysis in which the problem solver consistently compares the current problem state with the goal state (Greeno, 1978).

The implications of this type of problem-solving research for instructional design are limited. "Puzzle problems have been studied largely because they are not complicated by requiring background knowledge, and because they reveal the strategies that people employ in searching for a solution" (Chi & Glaser, 1985, p. 228). They are domain-independent and not tied either to school practice or to real-world practice. Therefore the results of that research, as well as the problems themselves, are of especially limited usefulness for guiding the design of problem-solving instruction, because, in addition to being domain-independent, puzzles are also very constrained. That is, all elements and processes required for a solution are knowable and known (Kitchner, 1983). There is one, correct solution, and that solution is guaranteed by using a specific procedure. All other procedures are less efficient and therefore incorrect. These attributes are inconsistent with the nature of most situated, real-world problem solving. While puzzle problems are interesting test-beds for research, they are not relevant either to school learning or everyday practice and so will not be treated further in this paper.

Well-Structured Problems

The most commonly encountered problems, especially in schools and universities, are well-structured problems. Typically found at the end of textbook chapters, these well-structured *application problems* require the application of a finite number of concepts, rules, and principles being studied to a constrained problem situation. These problems have also been referred to as *transformation problems* (Greeno, 1978) which consist of a well-defined initial state, a known goal state, and constrained set of logical operators. These problems:

- Present all elements of the problem,
- Are presented to learners as well-defined problems with a probable solution (the parameters of problem specified in problem statement),
- Engage the application of a limited number of rules and principles that are organized in a predictive and prescriptive arrangement with well-defined, constrained parameters,
- Involve concepts and rules that appear regular and well-structured in a domain of knowledge that also appears well-structured and predictable,
- Possess correct, convergent answers,
- Possess knowable, comprehensible solutions where the relationship between decision choices and all problem states is known or probabilistic (Wood, 1983), and
- Have a preferred, prescribed solution process.

These problems are more domain- or content-dependent than puzzle problems But they call on skills that can be transferred only to similar types of problems. Their solution is usually dependent upon a constrained knowledge base presented in the textbook chapter preceding the problem.

A primary purpose in distinguishing between well-structured and ill-structured problems results from the commonly held assumption that skills in solving well-structured, classroom problems will transfer positively to real world, situated, ill-structured problems. It is important to recognize that effects of well-structured problems in school contexts have limited relevance and transferability to solving problems that are situated in everyday contexts, which are described briefly next.

Ill-Structured Problems

Ill-structured problems are typically situated in and emergent from a specific context. In situated problems, one or more aspects of the problem situation are not well specified, the problem descriptions are not clear or well defined, or the information needed to solve them is not contained in the problem statement (Chi & Glaser, 1985). Ill-structured problems are the kinds of problems that are encountered in everyday practice, so they are typically emergent dilemmas. Because they are not constrained by the content domains being studied in classrooms, their solutions are not predictable or convergent. They may also require the integration of several content domains. Solutions to problems such as pollution may require components from math, science, political science, and psychology. There may be many alternative solutions to problems. However, because they are situated in everyday practice, they are much more interesting and meaningful to learners, who are required to define the problem and determine what information and skills are needed to help solve it. So, ill-structured problems:

- Appear ill-defined because one or more of the problem elements are unknown or not known with any degree of confidence (Wood, 1983),
- Have vaguely defined or unclear goals and unstated constraints (Voss, 1988),
- Possess multiple solutions, solution paths, or no solutions at all (Kitchner, 1983), that is, no consensual agreement on the appropriate solution,

- Possess multiple criteria for evaluating solutions,
- Possess less manipulable parameters,
- Have no prototypic cases because case elements are differentially important in different contexts and because they interact (Spiro, Vispoel, Schmitz, Samarapungavan, & Boerger, 1987; Spiro, Coulson, Feltovich, & Anderson, 1988),
- Present uncertainty about which concepts, rules, and principles are necessary for the solution or how they are organized,
- Possess relationships between concepts, rules, and principles that are inconsistent between cases,
- Offer no general rules or principles for describing or predicting most of the cases,
- Have no explicit means for determining appropriate action,
- Require learners to express personal opinions or beliefs about the problem, and are therefore uniquely human interpersonal activities (Meacham & Emont, 1989), and
- Require learners to make judgments about the problem and defend them.

Archetypal examples of ill-structured problems are instructional design problems (one of which will be used later to exemplify ill-structured problem solving). In most cases, the designer is constrained by circumstances, though in most design problems, there are a variety of solutions, each one of which may work as well as any other. Without empirical proof, the designer is required to make judgments about the situation and prescribe solutions based on them. Politics and sociology also offer a wealth of ill-structured problems, since few people ever agree on any particular political solution to any problem.

While early information processing theories of problems held 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), more recent research in situated and everyday problem solving (e.g., Lave, 1988) makes clear distinctions between convergent problem-solving thinking and the thinking required to solve everyday problems. Some preliminary research (Dunkle, Schraw, & Bendixen, 1995) has 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. Clearly more research is needed to substantiate this finding, yet it is obvious that well-structured and ill-structured problem solving engage different skills.

TOWARD AN INSTRUCTIONAL DESIGN MODEL FOR WELL-STRUCTURED PROBLEMS

Problem Solving Process for Well-Structured Problems

Information processing is the dominant theory in cognitive psychology that has scaffolded most theories of problem solving. Information processing models of problem solving, such as the classic General Problem Solver (Newell & Simon, 1972), generally specify two sets of thinking processes associated with the problem-solving processes, understanding processes and search processes. A popular problem-solving model that was derived from this earlier work is the IDEAL (identifying, defining, exploring, acting, and looking back) problem solver (Bransford & Stein, 1984). The IDEAL process for solving problems involves identifying potential problems, defining and representing the problem, exploring possible strategies, acting on those strategies, and looking back and evaluating the effects of those activities. Gick (1986) synthesized these and other problem solving models (Greeno, 1978) into a simplified schematic of the well-structured problem-solving process (Fig. 1).

The process begins with the learner generating a representation of the problem. What is the problem and the situation that gave rise to it? What kind of problem is it? This process involves problem decomposition and classification of the type of problem. Next, according to Gick's model, the problem solver searches for or generates possible solutions to the problem, which are then implemented and tested. Until a successful solution is found, the process continues by rerepresenting the problem or generating alternative solution hypotheses, which are then tested. There is no emphasis on finding more than one solution that will work, so when a successful solution is tested, the problem-solving process concludes. This automated view of well-structured problem solving supports the identification and implementation of the response that is deemed correct in terms of efficiency or accuracy.

Even well-structured problem solving is more complex than this overly simplified view based on information processing theory. For instance, the process of problem representation is better conceived as the creation of a problem space. This process involves mapping the problem statement onto prior knowledge and constructing a personal interpretation of the problem (i.e., problem space). In the problem space, the solver attempts to decompose the problem while identifying an appropriate solution state. These processes are dynamically related, using a means-ends analysis to interactively reconcile the problem with each potential solution. The problem-representation process may be scaffolded by presenting a conceptual model to the learner during the problem-representation process. Conceptual models illustrate the structural relationships among the problem components (Figure 2 presents a conceptual model of this problem-solving process). Each of the steps in this problem-solving process will now be examined in more detail.

Step 1: Problem Representation

The first step in well-structured problem solving is understanding the task, that is, extracting from the problem statement what the goal is. "What do I need to produce here; what is an acceptable solution going to look like?" Simultaneously, problem solvers isolate attributes of the problem. Problem solvers first attempt to represent the problem mentally by decomposing the problem statement and mapping the problem onto prior knowledge (as represented in Fig. 2). That process produces the learner's problem representation (problem space) which includes the problem solver's "understanding of the givens, the goal, the underlying structure of the possible solutions, and any problem solving strategies that can be used to solve this task (Polson & Jeffries, 1985). The solution to any problem results from a search through this mental problem space, which accesses prior domain knowledge and the hypothesis-generating and solution-finding processes required to act on the problem. It is important to note that problem representations are constructed by individuals in response to a problem-solving task that is presented, not one that emerges from the context or one that they generate themselves.

Representing the problem intentionally links the problem to existing knowledge. This process is known as schema activation, and what the learner is seeking is a schema for solving that particular type of problem. If the learner possesses a complete schema for that problem type, then the problem statement can be easily mapped onto the existing problem schema. Existing problem schemas result from previous experience in solving the particular type of problem being posed, enabling the learner to proceed directly to the implementation stage of problem solving (Gick, 1986) and try out the activated solution. Experts are bet-

Figure 1 ☐ Simplified Schematic of Problem-Solving Process (Gick, 1986).

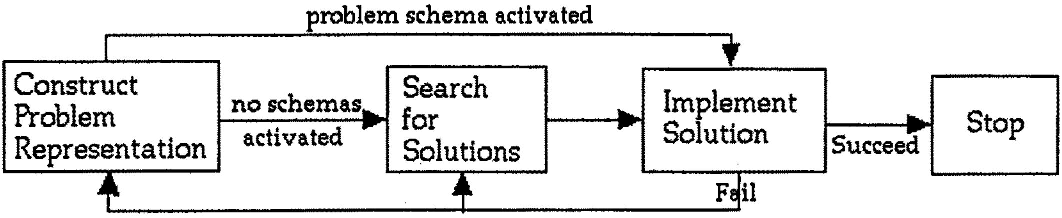
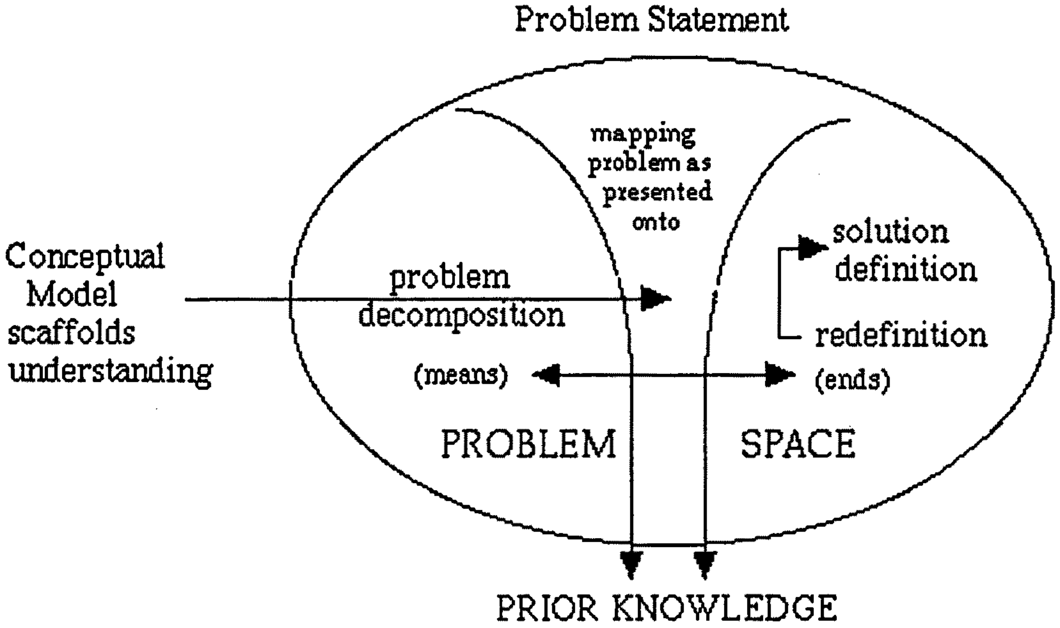


Figure 2 ☐ Conceptual Model of the Well-Structured Problem-Solving Process.



ter problem solvers because they recognize different problem states which invoke certain solutions (Sweller, 1988). If the type of problem is recognized, then little searching through the problem space is required. Novices, who do not possess problem schemas, are not able to recognize problem types, so they must rely on general problem-solving strategies, such as means-ends analysis, which takes them to the second stage in order to search for solutions.

Step 2: Search for Solutions

The key to problem solving is an adequate problem representation. Search processes by inexperienced or lower-ability learners tend to be haphazard and incoherent. A variety of strategies to support the search for solutions have been recommended by researchers. Most are heuristic strategies that require considerable skills on the part of the problem solver, which is uncertain, since most novice problem solvers are novices because they lack these strategies and problem schemas. One or more of the following strategies need to be engaged by instruction designed to facilitate problem solving.

Recall Analogical Problems. Recalling a previously solved problem and applying that solution method to a current problem is a very natural step in problem solving; usually the first method that people use according to Polya (1957). When faced with a problem, we naturally ask ourselves if we have experienced a similar problem. Using analogical problems requires that learners recognize the similarity between the previous and current problems and that the learner can recall the solution method used in the previous problem (Reed, 1992). In one of the most famous studies on analogical problem solving, Gick and Holyoak (1980) studied whether learners could analogize a dispersion solution to a radiation problem in medicine. Learners were required to map the solution to a military problem onto the medical problem. When prompted to think about the previous problem, most learners successfully applied the solution strategy to the medical problem. However, when the prompt to think about the previous problem was withheld, mapping of the previous problem decreased. Learners can use analogous problems when prompted to, but may not if unprompted. Recalling analogical problems is probably the first strategy that should be tried by learners and supported by instruction. If the learner cannot recall a similar problem or cannot apply an analogical problem, then other solution-generating strategies may be used.

Means-Ends Analysis. Means-ends analysis involves reducing the discrepancy between the current state and the goal state of the problem by applying problem-solving operators (Gick, 1986). It was first articulated in the General Problem Solver Model (Ernst & Newell, 1969) where the problem solver isolates the goals to be achieved and then systematically selects the methods (means) to achieve each of those goals. Having isolated the goals, the solver selects the most important difference and then selects a means to reduce that discrepancy, proceeding to the next most important difference until a complete problem solution is developed. For example, if the goal is to construct a wheelbarrow, then the builder would apply operators (e.g., visit the building materials store to acquire a wheel, brackets, wood and other required materials). Means-end analysis is a recursive process which identifies discrepancies (e.g., what materials are required), which in turn require planning to reduce those discrepancies.

The disadvantage of means-ends analysis is that it impedes schema acquisition, that is, focusing on the attributes of the problem in order to better classify the kind of problem that exists (Sweller, 1988). In order to become effective means-ends problem solvers, it is necessary to focus selectively on aspects of the problem. Reflecting on how previous problems were solved, that is, associating problem states with categories of problem solutions, requires processing capacity which is interfered with by the cognitive load imposed by goal-oriented strategies such as means-ends analysis.

Decomposing and Simplifying: Finding Sub-Goals. Breaking a problem down into subproblems is a generalized strategy that has been often recommended (Polson & Jeffries, 1985). In this strategy, the learner divides the problem into smaller subproblems and then applies the decomposition process to the subproblems until they are small enough to suggest an obvious solution. If the learner knows about a subgoal state that can be reached in fewer steps, then the possible number of solution paths is reduced, making the problem easier to solve. However, there is little cogent advice available on how this heuristic can be applied to actual problems. Decomposition, like most of these general strategies, requires that the learner have complete knowledge of the techniques and problem-solving domains (Polson & Jeffries, 1985). Because of this limitation, more prescriptive methods for supporting novice problem solvers will be presented in the instructional design model described later.

Generate/Test. The least structured and therefore weakest of the solution-generating methods is the generate-and-test method. Essentially, the problem solver brainstorms possible solutions which are then evaluated for their potential to solve the problem. This is perhaps the most common method for untrained problem solvers and relies on the general, intellectual abilities of the person generating the solutions. Therefore, it cannot be recommended as an instructional design strategy.

Step 3: Implement Solutions

The final step in the well-structured problem-solving process is trying out the solutions that the learner has generated. This is often an iterative process of testing the procedures contained in their problem schemas. If the solution works, the problem is solved. If the solution fails to work, then the learner should generate a new hypothesis or adjust the process to yield another answer. But, identifying clues from the failed attempts and using them to generate new solutions is difficult, especially for learners unaccustomed to failures. Learners will need a lot of coaching during this process, including motivational coaching to keep trying along with prompting of aspects of the failed attempts that can be used to generate new solutions. Often, what learners are trying out are formulas that they have recalled and related to specific problem conditions.

Learners may also be trying out hypotheses that are appropriately grounded in the problem space and problem elements.

Designing and Developing Well-Structured Problem-Solving Instruction

Using the problem-solving process depicted in Figures 1 and 2 as the model for how well-structured problems are solved, this section describes an instructional design blueprint for supporting that process. This process contains most of the instructional components suggested by other models of instructional design, including examples and practice. A lean, representative example of a well-structured problem solution can be found in the Appendix.

Step 1: Review Prerequisite Component Concepts, Rules, and Principles

Solving well-structured problems requires learners to identify, select, and apply relevant domain information. The concepts, principles, and procedures that are required to solve a problem (the component skills necessary to build their problem schema) should be reviewed or presented as concept and rule lessons prior to beginning the problem-solving lesson. (Understanding of concepts such as heterozygous, homozygous, dominant, and recessive and the rule involving Mendelian Genetics and the PUNNET Square are needed to solve genetics problems illustrated in the Appendix.) Alternatively, learners' comprehension of these problem components may be pretested, or assistance may be embedded in the problem-solving lesson as help, job aids, or performance support to be accessed by the learner.

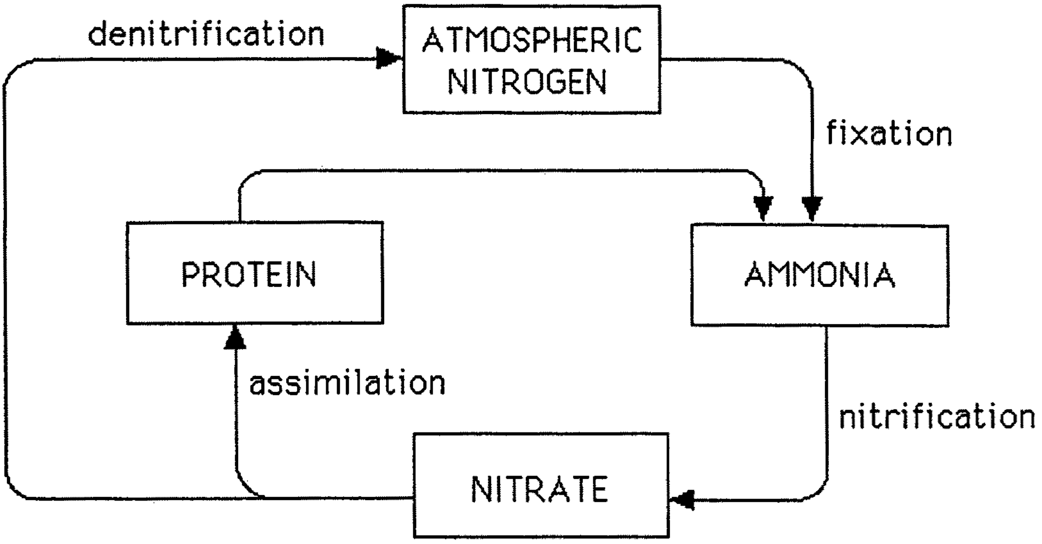
Step 2: Present Conceptual or Causal Model of Problem Domain

An effective means for helping learners construct appropriate problem representations (i.e., develop an appropriate problem schema) is to provide a graphic organizer of the problem domain. These diagrams can assume different forms. The best researched form of conceptual organizer is the conceptual model (Mayer, 1989). A good conceptual model contains a visual representation of all of the essential parts, states, or actions encountered in the problem and the relationships between them at a level of detail and familiarity that are appropriate for the learners. Having provided conceptual models in problem solving lessons in BASIC computer language, the camera, database systems, physics, and others, Mayer (1989) concluded that providing concrete, conceptual models for learners improves conceptual retention, reduces verbatim recall, and improves problem-solving transfer. Figure 3 illustrates a conceptual model for facilitating understanding of the nitrogen cycle, just as Figure 2 illustrates a conceptual model for the problem-solving process. The sample well-structured problem presented in the Appendix also illustrates a conceptual model for solving genetics problems, consisting of parent flies and their potential offspring.

Conceptual models must correspond to the events and objects that they represent. Good models make intuitive sense to the learners, because they are transparent to the learners and use vocabulary and concepts that are appropriate for the learners. The reason for intentionally illustrating the conceptual components in the problem space is to enhance learners' mental models of the content being studied. Conceptual models can be effectively presented before instruction or during instruction .

Another conceptual reason for providing learners with conceptual models of the problem domain being studied is because they explicitly represent the structural knowledge (Jonassen, Beissner, & Yacci, 1993) required to support problem solving. Robertson (1990) found that the extent to which the learner's cognitive structures contained relevant structural knowledge was a strong predictor of how well learners would solve transfer problems in physics. In fact, structural knowledge was a much stronger predictor than either aptitude (as measured by standardized test scores) or performance on a set of similar problems,

Figure 3 ☐ Conceptual Model for Understanding Nitrogen Cycle (Mayer, 1989).



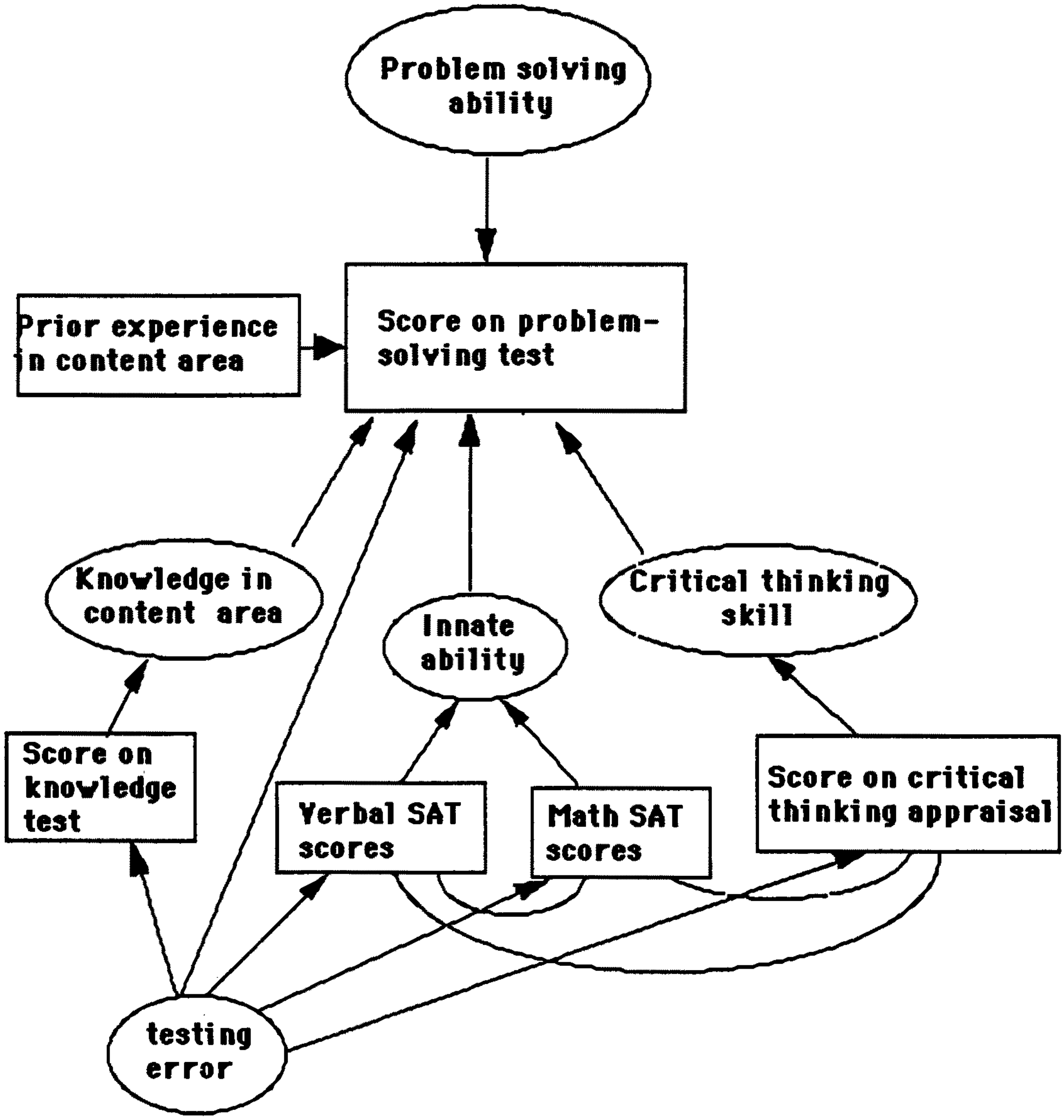
which is the underlying assumption of the analogical problem strategy. The similarity of the learners' underlying cognitive structure with the expert is highly predictive of problem-solving ability (Gordon & Gill, 1989). So, cognitive structures that connect important concepts in the knowledge base are important to understanding the domain, which is essential to constructing a meaningful problem representation. These concept maps convey important structural knowledge, which helps learners to build appropriate domain-specific problem representations.

There are a number of alternative graphic representations of structural knowledge that may also support the problem representation process. An alternative representation to conceptual models that is likely (though not yet empirically tested) to support well-structured problem solving is the causal interaction map. Causal interaction maps (Figure 4) show causal and correlational relationships between both observed and unobserved variables. With causal interaction maps, the interactions between variables that lead to a given effect are made explicit in a graphic format. Understanding these causal links between the attributes of a problem can support the search for solutions to the problem. Causal interaction maps more closely represent the solution process than conceptual models, which are designed to represent domain knowledge.

Step 3: Model Problem Solving Performance in Worked Examples

The purpose of worked examples is to model required problem-solving performance, including both a description by an experienced (though preferably not an expert) problem solver of how the problems are solved as well as the thought processes that are engaged by the problem-solving experience. Worked examples help learners to construct useful problem schemas. They can help learners categorize problems with similar solutions and construct solutions to novel problems by analogy to the example (Anderson, Farrell, & Sauers, 1984; Sweller & Cooper, 1985). In fact, considerable research has shown that problem-solving performance improved more after studying as few as two worked examples than from solving several well-structured problems (Cooper & Sweller, 1987; Sweller & Cooper, 1985; Ward & Sweller, 1990) . The well-structured problem in the Appendix includes a worked example of a genetics problem which employs the Punnet Square. This solution

Figure 4 ☐ Causal interaction map.



shows not only the process used for solving the problem but also some of the reflective thinking that is essential to that process.

Developing worked examples is difficult. Since problem solvers need to recognize the initial problem conditions, attention in the worked examples should be directed to differences in the problem states. However, worked examples that require learners to split their attention between multiple sources of information, such as problem statements, diagrams, and formulae, and then integrate those sources, are ineffective (Ward & Sweller, 1990). They increase cognitive load. So, the conceptual model should probably not be included in the worked example. The worked example in the Appendix demonstrates the reasoning necessary to use the Punnet Square in order to make predictions about offspring. Worked examples should model for the learner how they should be constructing problem representations.

While worked examples have been shown to be an efficient method for supporting novice learners, there are questions about transferability of problem-solving skills acquired from passively viewing worked examples. Charney, Reder, and Kusbit (1990) noted motivational problems resulting from the relative passivity of worked examples. Learners accustomed to explicit guidance and direction may be less likely to set, and work towards, their own problem goals.

In order to evaluate this issue, researchers have compared worked examples with more exploratory approaches, finding that learning by trial and error takes more time than rule-based learning, but that it promotes transfer of learning to new tasks (Scandura, 1964). Learners may spend time fruitlessly on incorrect solution paths and therefore may fail to acquire good models of solutions (Sweller and Cooper, 1985). Several comparison studies of worked examples and trial-and-error problem solving in the domain of algebra showed that studying worked examples was more beneficial since the standard means-end problem-solving strategies used by novices impose a heavier cognitive load than worked examples (Sweller, 1988; Tarmizi & Sweller, 1988). Experienced problem solvers possess a variety of problem-solving scripts that they apply to new problem situations. Jonassen, Doricott, and Engels (1995) found that students receiving worked examples solved related fluid dynamics problems more effectively and efficiently while learners in the exploratory group solved more far-transfer problems. Exploration requires learners to identify and test more hypotheses during problem solving, so it is reasonable to expect them to be able to transfer that performance to an unrelated performance.

Worked examples should probably not be developed using experts to model the process. Rather, it is important to use a journeyman as a model, someone who is competent and experienced in solving this type of problem, but someone who can still articulate all aspects of the problem state and solution. Experts' representations of problems include solutions and situations in which the problems occur, so they rarely have to search their problem space for solutions, as do novices (Gick, 1986). Novices' problem schemas are based on superficial similarity among problems, whereas experts' schemas are based on solution principles (Chi, Feltovich, & Glaser, 1981). Experts use a "working forward" strategy (working forward from given information toward the goal by choosing equations that contain the givens and solving the goal) while novices "work backward" (selecting equations that contain the goal and trying to plug in the givens) (Simon & Simon, 1978). Finally, experts' schemas contain physical descriptions of the situations, where novices describe problems only in terms of equations. All of these differences suggest that novice learners will be unable to understand experts' solutions. However, some of these elements, particularly relating the problem to physical, real-world phenomena, are useful to include in the model. Have the journeyman think aloud as s/he solves the problem. In addition to articulating procedures, it is important that the model-problem solver also attempt to model the use of solution strategies, such as means-end, decomposition, diagramming problem elements, and using analogous problems. Be sure to include this information in the worked examples.

Step 4: Present Practice Problems

Learning from worked examples may help learners form appropriate representations of concepts and problem situations in the domain. However, worked examples alone may not provide sufficient integration of developing and testing solutions as learners would achieve with meaningful practice. Problem schemas can develop rather quickly in learners, however automating the rule sequences for solving them is a slow process that requires extensive practice (Cooper & Sweller, 1987; Sweller & Cooper, 1985). Worked examples affect the acquisition of problem schemas first and only later improve rule automation. So, the combination of worked examples plus extended practice is most likely to facilitate the acquisition of problem schemas and the transfer of those schemas to novel problems.

Present the practice problems to the learner in the form in which they will be assessed (see sample genetics practice problem in the well-structured problem in the Appendix). If the tests employ word problems, then be sure to provide word problems during practice. Be certain that all of the problem elements necessary to solve it are presented to the learner, or ensure that the learner knows where and how to find them. Problems are often made more realistic by withholding some problem elements, requiring that learners identify missing information that is needed. This is appropriate and effective so long as learners know that critical information may be missing from the problem and where and how to retrieve the information necessary to solve it. If learners are not aware that information may be missing, the problem will surely result in inappropriate problem schemas that may transfer to other problems as well as a lot of frustration on the part of the learners. Problems are also made realistic by including information in the problem statement that is not required in its solution. Disambiguating relevant from irrelevant information is an important problem-solving skill and so should be practiced. Again, the inclusion of irrelevant problem information should be consistent between worked examples (described next), practice problems, and assessment problems.

Step 5: Support the Search for Solutions

Having helped the learners to construct meaningful problem representations, it is now appropriate to provide supports that help learners to find and try different solutions. The following supports may be made available to learners to assist them in generating and testing plausible solutions.

One approach is to provide analogical problems. These problems provide synopses of similar problems that others have previously solved. Jonassen, Ambruso, and Olesen (1992) provided learners access to a rational set of 24 cases that were linked to the case being solved. These cases were made available to the learners as closely related cases, somewhat related, or unrelated to the case being solved, based on the number of indexical links in common. Analogical cases are powerful scaffolds, because they supplant prior experiences and problem spaces not possessed by learners. Make the problem elements of the analogical problem obvious in order to assist learners in mapping the previous problem onto the new one.

Another support strategy is to provide advice or hints on breaking down the problem into subproblems (means/ends) that can be more easily solved by highlighting relevant cues or providing a solution template. Prompts may also be given on operators or actions that can be taken to solve the subproblems.

Finally, it is essential, as with any form of practice during instruction, to provide adequate feedback about learners' attempts to solve the problem. Feedback should constitute more than simple knowledge of results (correct/incorrect) or correct answer feedback. In addition to apprising the learners of their correctness, it is important, if their answers are incorrect, to determine where the problem-solving process went wrong and provide either coaching or the correct solution process from that point in the problem. Feedback should also address the conceptual assumptions (possibly referring to the conceptual model) and methods used to solve the problem. Misconceptions and solution errors are often directly linked to inappropriate assumptions.

These instructional design interventions, problem diagrams and concept maps, worked examples, and analogical problems, are all problem-solving scaffolds. Scaffolds are temporary frameworks to support any kind of learning. They support learners in their "zone of proximal development" to perform complex tasks, such as problem solving, with help where, without help, they would be unable to perform. These scaffolds should be faded out as soon as possible. That is, they should not be made consistently available to learners. Provide students the opportunity to use these scaffolds in arriving at a solution. Allow them to test their solutions.

Step 6: Reflect on Problem State and Problem Solution

Since the cognitive load of problem solving interferes with the acquisition of appropriate problem schemas (Sweller, 1988), learners should reflect on initial problem conditions in order to facilitate the acquisition of relevant problem schemas. Learners should note the characteristics of the problem as presented: the situation, the knowns and unknowns, and the problem as stated. They should then reflect on the solution processes that were most effective and ineffective in solving the problem. Learners can even create tables or databases of problem types and solutions. Developing strong associations between the type of problem encountered and the types of solutions used is very likely to help learners to develop stronger problem schemas which will help them to become better problem solvers in the future. Questions regarding the initial conditions and how they related to the solution are not presented in the very lean well-structured problem in the Appendix but rather are included in the worked example. Asking questions requiring learners to compare initial problem conditions with solutions can only help to solidify the learning.

TOWARD AN INSTRUCTIONAL DESIGN MODEL FOR ILL-STRUCTURED PROBLEMS

For the past two decades, problem solving, like most of cognitive psychology, has moved slowly away from information-processing theory as its conceptual base. Polson and Jeffries (1985) analyzed four problem-solving approaches that proposed a divergent production paradigm rather than the convergent production that describes well-structured problems. These approaches viewed thinking as a skill but diverged from traditional problem-solving models by emphasizing perception and pattern recognition as well as divergent and creative thinking in order to generate as many alternative representations of the problem as possible (Polson & Jeffries, 1985). Problem solving programs, such as the Productive THINKING Program (Covington, Crutchfield, Davies, & Olton, 1974), CoRT Thinking Materials (deBono, 1974), Patterns of Problem Solving (Rubenstein, 1975), and How to Solve Problems (Wickelgren, 1973) sought to teach replicable, general, domain-independent, perceptually oriented problem-solving skills with the expectation that they would work with ill-structured problems. However, two major limitations of divergent production approaches became apparent: students experienced difficulty in fluently generating alternative solutions and representations, and the methods did not transfer to solving ill-structured problems.

Process for Solving Ill-Structured Problems

The remainder of this paper focuses on solving ill-structured problems, that is, ill-defined problems situated in the real world. Ill-structured problems are those on which opposing or contradictory evidence and opinions exist, for which there is not a single, correct solution that can be determined by employing a specific decision-making process (Kitchner, 1983). Classical ill-structured problems exist in international relations (Voss, Wolfe, Lawrence, & Engle, 1991), 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, 1989, p. 273). International relations problems involve decision making, and decision making is grossly affected by the political context in which those decisions are made. Ill-structured problems seldom have a single, best solution. Rather, they typically have several possible solutions, each of which offers advantages and disadvantages to different people and situations in the context of their application. For instance, selecting a route for a new expressway through a metropolitan area typically engenders strong and opposing views about different locations. No solution will please all of the constituents or maximize the goal of carrying traffic, given fiscal and political constraints. The question itself, whether an expressway should even be built, is often challenged by environmental groups and historians as being an appropriate solution.

Conceptually, ill-structured problem solving may be thought of as a design process, not a systematic search for problem solutions (Schön, 1990). It is a frame experiment in which the problem solver engages in a reflective conversation with the elements of the problem situation. This dialectic process is illustrated in think-aloud protocols collected by Sinnott (1989) while adults attempted to solve a variety of everyday problems. As designers, when we frame a situation we create an initial design structure within which we begin to invent and implement solutions. As with well-structured problems, sources of data for generating that design structure may include analogical problems and stories. However, in this model, the problem solvers must frame the design problem, recognize the divergent perspectives, collect evidence to support or reject the alternative proposals and ultimately synthesize their own understanding of the situation rather than find a solution for a prescribed problem.

Sinnott (1989) used think-aloud protocols to induce a model for solving ill-structured problems. She discovered a series of dialectic processes for constructing the problem space, processes for generating and choosing solutions, and a variety of memory, monitoring, and non-cognitive strategies for supporting problem solving. These will be addressed in the description of the ill-structured problem-solving process that follows.

Step 1: Learners Articulate Problem Space and Contextual Constraints

The first step in the problem-solving process is to decide if a problem really exists. Many ill-structured problems are in fact pseudo-problems. They may appear to have an unknown when, in reality, there is a hidden known, and so there is no problem. Next, the problem solver must determine what the nature of the problem is. Ill-structured problems are ill-structured because there may be multiple representations or understandings of the problem. So, identifying an appropriate problem space from among the competing options is perhaps the most important part of ill-structured problem solving. LeBlanc and Fogler (1995) provide numerous examples of this type of ill-defined problem. For example, when guests on the upper floors of a hotel complained about slow elevators, searching for a way to speed up the elevators masked the real problem of taking the guests' minds off their wait, so management installed mirrors in front of the elevators, thus eliminating the customer complaints and solving the problem from the hotel's perspective. Situated, real world problems are emergent, not predefined. So, the solver must examine the context from which the problem emerged and determine what the nature of the problem is.

Therefore, ill-structured problems are said to be domain dependent or context dependent because they require the problem solver to think about the problems as realistic situations rather than to rely on information from the chapter in constrained problem representations (Bransford, 1994). Problem solving requires access to well-organized, domain-specific knowledge, leading Bransford to ask, "Who ya gonna call?" if your dog is misbehaving—a dog trainer, a plumber, or a brain surgeon? He also argues that expertise and wisdom cannot be taught; they are acquired through experience. Meaningful problem solving is developed through experience in solving problems, not from canned problems with convergent answers.

How do we conceive of wisdom and domain knowledge? Chi, Glaser & Rees (1982) describe them in terms of the problem schemas. Experts possess more highly developed problem schemas because they represent problems physically in terms of real world mechanisms, which makes the problem more meaningful, easier to check for errors, and easier to define (Chi et al., 1982). Good problem solvers recognize the type of problem and organize their knowledge in problem-centered ways, while poor problem solvers focus on surface characteristics of the problem (de Jong & Ferguson-Hessler, 1986) that are inadequate for solving the problem. Therefore, rules and concepts are accessed and applied to problem situations more quickly. Domain knowledge and problem-solving skill develop from experience in solving problems. If your automobile is broken, will you call a brain surgeon? Expertise in diagnosing automotive problems comes from years of experience and the contexts of thousands of cars. That is why experts solve problems in one fourth the time of novices (Simon & Simon, 1978) and why they commit significantly fewer errors.

As with well-structured problems, constructing the problem space is perhaps the most important activity in solving ill-structured problems. However, rather than recognizing and classifying problem types (e.g., a kinetics problem), ill-structured problems require that learners assemble a large amount of relevant, problem-related information from memory (Voss & Post, 1989). Learners cannot retrieve the appropriate rules from the chapter(s) being studied. Ill-structured problems engage a broader range of conceptual knowledge about the problem domain. Ill-structured problems cannot be solved by applying a constrained set of rules. They also require that learners construct a problem space that contains all of the possible states of the problem, the problem operators, and the problem constraints (Voss, 1988). So, an important strategy for the problem solver is to examine all of the possible causes of the problem as well as the constraints.

Ill-structured problems interact with and so are constrained by contextual factors. International relations problems, for example, are constrained by incomplete, inaccurate, or ambiguous information (e.g., speeches require interpretation), heavy processing loads entailed by a lot of information, and often several, divergent goals that must be taken into account (Voss et al., 1991).

An important metacognitive strategy that individuals should apply is to reflect on what they know about a problem domain. Learners must answer questions, such as: How much do I know about this problem and its domain? What do I believe to be true about it; what are my biases? Have I heard stories or accounts about this situation? Have I read anything about it? Do I know where I might find information about it? If not, then who might? Clearly, better developed domain knowledge (prior knowledge) will enhance problem-solving ability in any particular domain. Learners must learn how to relate problem aspects to their own personal knowledge. This will likely be accomplished through modeling this performance or a set of procedural prompts by the teacher to review how much is already known about the problem.

Step 2: Identify and Clarify Alternative Opinions, Positions, and Perspectives of Stakeholders

Ill-structured problems are dialectical in nature, requiring the problem solver to reconcile conflicting conceptualizations of the problem (Churchman, 1971). So rather than allowing the construction of a single problem space, ill-structured problems may require the learner to construct multiple problem spaces. The problem solver must then traverse the cognitive or affective associations between problem spaces in order to decide which problem schema is most relevant and useful for solving the problem (Sinnott, 1989). Selecting a problem space for ill-structured problems necessarily involves identifying alternative views or perspectives on the problem. Who are the stakeholders in the problem situation, and what are their goals? How do they perceive the nature of the problem? In order to comprehend the complexity of the problem, learners must perceive and reconcile different interpretations of phenomena involved. "It is only through the use of multiple schemata, concepts, and thematic perspectives that the multi-faceted nature of the content area can be represented and appreciated" (Jacobson, 1990, p. 21). Having identified the problems and the goals that different people have, it is important for the problem solver to identify all of the various perspectives, views, and opinions on that problem because ill-structured problems usually have divergent or alternative solutions. Ill-structured problem solving is a process of reflective judgment (Kitchner & King, 1981) in which learners reconcile the uncertainty of knowledge through the process of inquiry into their beliefs.

Step 3: Generate Possible Problem Solutions

This working-forward process of identifying positions and inferring solutions from them is consistent with the way that experts solve problems. While most instructional systems recommend articulating goals before means, it is often more natural in ill-structured problem situations to identify the various positions first and then infer how people who hold that position would solve the problem. Identifying solution states by analyzing possible causes of the problem focuses the solution-generation process on solutions that alleviate the causes. The solution alternatives that are generated are a function of the characteristics and constraints of the problem representation (Voss et al., 1991). That is, ill-structured problems possess multiple solutions because there are multiple representations of the problem. The problem solver's perceptions of problem constraints are the primary factors that determine which alternative is selected. Different problem representations of any case lead to alternative solutions, so considering alternatives and evaluating their outcomes further constrains the solution. The process of generating solutions is a creative process that relies not only on prior experiences but also unrelated thoughts and emotions (Sinnott, 1989). Solvers select solutions that they know are reachable and known to the solver. In essence, the learners are building their own mental model of the problem which enables them to identify and select or synthesize a solution. This process requires epistemic knowledge about the validity of alternative solutions (Kitchner, 1983) which is addressed in the next step.

Step 4: Assess the Viability of Alternative Solutions by Constructing Arguments and Articulating Personal Beliefs

Since ill-structured problems typically do not have a single, best solution, a learner's representation of it should assume the form of an argument for a preferred solution or against alternative solutions. This construction is facilitated by clarifying just how others holding alternative views would argue and then agreeing or disagreeing with the attributes of their arguments. In so doing, the learner is constructing his/her own arguments and developing a personal position statement about a preferred solution. The resulting mental model of the problem will support the learner's decision and justify the chosen solution. Learners should select or construct a solution that will be viable, reflect on how they came to that decision, and learn to justify that solution. The "best" solution is the one that is most viable, that is, most defensible; the one for which the learner can provide the most cogent argument.

The learner needs to gather evidence to support or reject various perspectives and to support any arguments made for one or another. Solving ill-structured problems, especially international relations problems, requires that learners develop cogent arguments in support of their solutions. The learner must make claims about the probable effects of events, objects, or phenomena on others, warrant those claims, and back them up with supportive statements, facts, or conjectures (Voss, 1988). By arguing and counterarguing (with themselves or in a group), learners are refining their problem representations and agreeing on the best course of action. Ill-structured problem solving becomes a process of iteratively restricting alternatives and refining arguments before selecting a solution.

Step 5: Monitor the Problem Space and Solution Options

A primary difference between ill-structured and well-structured problem solving is epistemic monitoring because good problem solvers "show more executive control of their own cognitive initiative" and "regulate their own thinking in a manner which is marked by more intensive . . . information processing" (Kluwe & Friedrichsen, 1985, p. 207). It is necessary for all kinds of problem solving to make a problem-solving plan intentionally and to carry out that plan. Planning is an essential executive strategy and provides evidence of metacognition. However, ill-structured problem solving should engage meta-metacognitive processes whereby individuals monitor the epistemic nature of the problems they are solving and the truth value of alternative solutions (Kitchner, 1983), not just the comprehension-monitoring metacognitive strategies that serve well-structured problem solving. This includes individuals' knowledge about the limits of knowing, the certainty of their knowledge, and their criteria for knowing. Epistemic cognition, according to Kitchner, "leads one to interpret the nature of the problem and to define the limits of any strategy to solving it." From this perspective, in order to solve a problem, the problem solver must first decide if the problem is solvable and whether there exist strategies or processes for solving it.

Ill-structured problem solving is more dialectical than well-structured problem solving. That is evident in the ill-structured instructional design problem presented in the Appendix and discussed in the following sections. Problem solving is a conversation where designers frame the problem they are facing in different ways (Schön, 1990). In doing so, they learn a repertoire of types, images, and metaphors that can be used to help them frame different dilemmas. So, problem solvers must have some epistemic knowledge about alternative solutions and then develop a strategy for representing or framing the problem and selecting or synthesizing a unique solution. It is important to note that the kind of epistemic monitoring described in this step occurs throughout Steps 1–4 and not as a separate, post hoc reflective process.

This monitoring process relies on a variety of memories: idiosyncratic memories, including personal histories such as school performance; emotional memories; problem-related memories; and even abstract rules (Sinnott, 1989). The memories that are accessed are generally controlled by the problem space. The richest are the personal memories related to prior problem-solving endeavors. These may support or impede attempts to generate solutions though. Learners who believe that they are good at mathematics are more likely to generate better solutions. While ill-structured problem solving is believed to be a contextually-driven process, learners may also retrieve and apply abstract rules, as with well-structured problems. Recent research (Kosonen & Winne, 1995) has shown that students who are explicitly taught abstract rules about statistics applied those rules across three different kinds of problems, despite common wisdom that abstract principles do not transfer well, especially in ill-defined problems.

Monitoring ill-structured problem-solving performance is a complex process where learners reflect not only on what they know and have been taught but also on what it means. Yet they must also go beyond what they know and believe to consider what others believe and to develop arguments to support their mental model of the problem space. In ill-structured problem solving, this model is emergent and dynamic, unlike the restricted problem schemas that define well-structured problems.

Problem-solving learning, especially in formal educational contexts, often ends here. Many problems are so complex and inaccessible that a recommended solution cannot be tried out, so it is sufficient merely to articulate the possible solutions and arguments. For instance, if you charged learners with trying to recommend a solution for the Bosnian political crisis, it would be impossible to try out their solutions. However, working through the problem construction and solution generation and its justification will doubtlessly engage learners in higher-order, problem-solving learning. In real-world contexts, problem solvers would necessarily have to try out their solutions.

Step 6: Implement and Monitor the Solution

Since ill-structured problems do not necessarily have a "correct" solution, the effectiveness of any solution can be determined only by how it performs. Following implementation of a solution, learners must monitor performance of the elements in the problem to see how they perform. How persuasive is the performance? Does it produce an acceptable solution to the involved parties? Is that solution satisfactory within the problem constraints articulated in the first step? Is it elegant and parsimonious? Could similar effects be achieved more efficiently or elegantly? Based on that performance, learners should potentially adapt both the solution and their mental model based upon performance (evidence of the effectiveness of the solution). Having tested various solutions and selected what the learners believe to be the most effective one, learners should then learn to make inferences about the utility of that solution for other problems. Drawing implications from their solution and extrapolating from the solution are essential to transferring the solution to other domain problems. The cognitive results of this stage are better integrated mental models of the problem space achieved by learners' reflecting on what they have learned.

Step 7: Adapt the Solution

If it is possible to try out the solution, then the problem-solving process would become an iterative process of monitoring and adapting the chosen solution based on feedback. Few problems are solved in only a single attempt. Problem solvers recommend a solution and then adjust and adapt it based on feedback.

Designing and Developing Ill-Structured Problem-Solving Instruction

In designing instruction that engages learners in ill-structured problem solving, the designer must work with subject matter experts and experienced practitioners to accomplish the following tasks. A sample ill-structured instructional design problem is described in the Appendix. It is a generic description of an ill-structured problem. Space constraints preclude presentation of all of the case material. In this performance-technology problem, there are multiple solutions. However, the more appropriate ones rely on discovering what kind of problem it is.

Step 1: Articulate Problem Context

Because ill-structured problems are more context-dependent than well-structured problems and because it will be necessary to develop an authentic task environment (the situational context of the problem) (Voss, 1988), it is necessary first to understand the context of the problem. Therefore, a context analysis needs to be conducted. What is the nature of the domain? What are the constraints imposed by the context? What kinds of problems are solved in this domain and, equally important, what are the contextual constraints that affect problems? In this case, malpractice suits are eroding profits and resulting in exorbitant premiums for the physicians. A legally binding method for insuring informed consent could result in millions of dollars in savings for this company and billions for the health industry.

Another reason for articulating the problem domain is that well-developed domain knowledge is essential to problem solving; at least as important as previous experience in solving problems in a particular domain (Robertson, 1990). Without adequate domain knowledge, even prior problem-solving skills cannot be transferred to a domain because learners do not transfer constructs from other domains to this one. Designers need to generate an inventory of all of the domain knowledge—not as a list of concepts, rules, and principles as with well-structured problems, but rather information about the context in which the problem is naturally embedded. What do people in this domain get paid to do? In this problem, the problem solver must understand not only complex medical procedures, but also important principles of tort law.

A potentially effective method for analyzing the task domain is activity theory. Activity theory (Leont'ev, 1978) assumes that "a minimal meaningful context for individual actions must be included in the basic unit of analysis" (Kuutti, 1996, p. 26) and so emphasizes the role of consciousness and the mediational roles of tools and sign systems in human activity. Human activity is more than the actions performed. Participating in an activity is performing conscious actions that have a goal. So, along with the actions that form activity structures, it is necessary to analyze the performers' conscious goals or intentions in the performance, the object of that performance, the language and tools they use, and the artifacts that they create. Activity theory is used extensively in designing computer systems (Bodker, 1991) and human-computer interfaces (Nardi, 1996).

The ill-structured case in the appendix is an instructional design problem involving patient education and assessment. Understanding the context and the roles of each entity (e.g., harried physicians, anxious patients, avaricious insurance company) are essential to a viable solution. The context will provide a considerable payoff for solving the problem.

Step 2: Introduce Problem Constraints

Instruction for well-structured problems would articulate the goals and solutions for the problem at this point. However, ill-structured problems seldom, if ever, have clear or obvious solutions or solution alternatives. What ill-structured problems do have are problem constraints or requirements that must be accommodated. However, a successful solution often must be implemented within a predetermined time frame and budget in order to meet the needs of the client. The solution must conform to certain environmental constraints which need to be introduced to the learners. The instruction supporting the medical problem in the Appendix must be delivered to confused and anxious patients, often in a hospital setting. There are also expectations about the effectiveness of the solution (in this case, to reduce court losses). The primary purpose of any intervention in this sample case is to show that the patient was adequately informed and clearly understood the risks inherent in the procedure. Most ill-structured problems have constraints that are imposed by a client or the situation. For instance, the commitment of nuclear weapons or massive armies are not acceptable solutions to the Bosnian crisis. The international political community would not accept or tolerate such solutions. So it is necessary to identify for the learners what requirements might reasonably constrain their solutions.

Step 3: Locate, Select, and Develop Cases for Learners

Having identified the skills needed by a practitioner, the next step is to select cases that necessarily engage those skills. The one case that has been introduced (it includes hundred of pages of support materials) is the cholecystectomy. According to cognitive flexibility theory (Spiro, Feltovich, Jacobson, & Coulson, 1992), anchored instruction (Cognition Technology Group at Vanderbilt, 1992), medical problem-solving environments (Jonassen, Mann, & Ambruso, 1996), the heart of instruction is the cases that include the contextualized problems that learners must solve. So, the designer must develop cases that represent probable real-world problems in the domain, that is, that are authentic. The obvious source of these cases is practitioners who can be interviewed. Anyone who has practiced in a domain for a significant length of time can identify a range of cases that involve problems to be solved. Insuring the relevance of the problem in the real world or its representativeness of the problem domain is essential to their success. The problems should be interesting and challenging, yet solvable. Assessing potential problems for use should include criteria such as realism (is the problem likely to be encountered in the real world?) the likelihood of different solutions and different opinions about the solution, and real-world criteria for evaluating the potential effectiveness of different solutions. Avoid the temptation of subject matter experts to include every problem aspect in a single mega-case. More, smaller cases promote transfer better than fewer, larger cases. The case in the appendix is one that designers are working on currently.

In order to ensure that learners are solving the real problem, consider performing a *causal analysis* of the problem (Jonassen et al., 1996). A causal analysis uses causal modeling theory to represent the information-processing requirements for connecting all antecedent-consequent actions in solving a case. That is, attempt to identify all of the possible solutions to the problems and then attempt to determine the probabilities of all of the possible causes of the problem. Causal modeling also provides a model for cognitively supporting the thinking required by the learners to diagnose and solve the problem. The requisite thinking can be scaffolded by the case environment (Jonassen, 1996). Space restrictions also prevent an elaboration of this process.

Step 4: Support Knowledge Base Construction

Another task analysis process applied to case-based learning entails identifying the alternative opinions and perspectives on the problem and instantiating those perspectives with a knowledge base of stories, accounts, reports, evidence, and information that pertains to that problem. Among the most powerful resources are stories by practitioners that relate the problem (Schank & Cleary, 1995). So, the designer in the ill-structured case in the Appendix needs to interview physicians and patients who have faced gall bladder surgery and capture their stories about problem aspects that they thought were relevant and how they went about solving the problem. They must also consult attorneys (plaintiff and defendant) as well as case law in order to identify positions, issues, and strategies that have been successful. It is important that these stories represent real and divergent perspectives. Reconciling multiple perspectives is perhaps the essence of ill-structured problem solving. Additional evidence, in the form of technical reports, video explanations, case law, and case histories should be collected and made available to learners in a simple way.

As stated before, learners must perceive and reconcile different interpretations of problem situations. In this example, opinions of medical staff, patients, family, and (of course) lawyers are important to the cases. In order to do so, learners need access to information related to those perspectives. So, identifying all of the constituents or stakeholders in a problem is important. Who has a meaningful perspective on the problem? What is their position? What stories can they tell to support their perspectives? What information sources are available to support these different perspectives? This evidence may be in the form of stories by concerned people, data, technical reports, newspaper and magazine reports, textbooks, Internet sources, or any other appropriate format. It is neither necessary nor appropriate to provide all of the information that learners need in order to solve a problem. It is reasonable to expect learners to search for some of the information. However, providing a structured knowledge base should scaffold their information collection.

A useful instructional model for conveying these multiple perspectives is cognitive flexibility theory (Spiro et al., 1987; Spiro et al., 1988). Cognitive flexibility theory conveys problem complexity by presenting multiple representations or thematic perspectives on the information. In order to construct useful knowledge structures, learners need to compare and contrast the similarities and differences between cases. Cognitive flexibility theory models an important characteristic of instruction for ill-structured problems—the provision of multiple perspectives and opinions.

Step 5: Support Argument Construction

As stated before, ill-structured problems are dialectical in nature, in which two or more opposing conceptualizations of the problem (different problem spaces) are used to support different arguments with opposing assumptions underlying them (Churchman, 1971). It is important that learners be able to articulate the differing assumptions in support of arguments for whatever solution that they recommend. The argument will provide the best evidence for domain knowledge that they have acquired. Developing cogent arguments to support divergent thinking [reflective judgment (Kitchner & King, 1981)] engages not only cognition and metacognition of the processes used to solve the problem but also epistemic cognition of the epistemic nature of the process and the truth or value of different solutions (Kitchner, 1983). In this case, it is also an important part of the problem. Understanding legal arguments is an essential part of the solution.

Getting learners to make reflective judgments about what can be known and what cannot is important to support in problem-solving instruction. That support may take the form of modeling the arguments for the solution to a related problem or prompting learners to reflect on what is known. If modeling is used, it is important that the perspectives of the different problem solvers (both the expert and the journeymen) be modeled for the learners. In the example in the Appendix, learners need to identify a design solution and justify that solution with specific reasoning. For example, "The problem is really an assessment problem. The goal is to certify the patient understands the risks inherent in surgery and accepts those risks. Therefore, the best solution is to . . . ." Modeling argumentation can also be scaffolded by providing an argument template or argument checklist. The arguments that are developed also provide a valuable assessment of the learner's problem-solving ability.

If coaching or prompting are used, the learners should be provided a series of reflective judgment prompts or questions (Kitchner & King, 1981), such as:

- Can you ever know for sure that your position is correct? Will we ever know which is the correct position?
- How did you come to hold that point of view? On what do you base it?
- When people differ about matters such as this, is it ever the case that one is right and the other wrong? One opinion worse and the other better?
- How is it possible that people can have such different points of view?
- What does it mean to you when the experts disagree on this issue?

The purpose of modeling and coaching is to engage the learner in considering each point of view and selecting the best one based on reasoning and evidence.

Step 6: Assess Problem Solutions

Assessing the solution of ill-structured problems is much more problematic than assessing well-structured problem solutions which have convergent, correct solutions that can be assessed as either right or wrong. Solutions to ill-structured problems are divergent and probabilistic. Evaluating learners' solutions must consider both process and product criteria. The product is the recommended solution. Trying out the solution to many realistic problems may be impossible in traditional classroom contexts, so we can only evaluate proposed solutions and their arguments. In most cases, solutions can only be evaluated in terms of their viability. The questions that we are most interested in answering are: Was the problem solved—did it (or is it likely to) go away? Was it solved within the constraints identified earlier? Can the learners articulate the causal relations implied by the solution to the problem, that is, can they explain why and how the problem was solved by their solution or why it was not solved by their solution? If the problems are authentic, as we hope they are, then other criteria for evaluating the viability of the solution emerge from the context. Will the solution please the client? Does it address all of the issues and constituents? Is it within budget and time parameters? In the case in the Appendix, having solutions evaluated by a judge or conducting moot court may provide meaningful assessments.

Second, the learners' problem-solving processes may also be evaluated. Did they accommodate important perspectives? How cogent was their argument for the proposed solution? Did the learners effectively reflect their own domain knowledge? What evidence did the learners provide that they thought deeply about the domain while solving the problem?

Implementing Ill-Structured Problems

Solving ill-structured problems is largely an iterative and cyclical process. Learners must adapt their strategies to the problems and the information they receive or generate. Designers will also have to adapt the nature of the resources they provide to learners. However, the process should generally proceed as illustrated in Table 1.

SUMMARY

Problem solving engages higher-order skills and is believed to be among the most authentic, relevant, and important skills that learners can develop. Models for well-structured and ill-structured problem solving have been presented along with models for designing and

Table 1 ☐ Implementation Process for Ill-Structured Problems

Designer/Developer	*Learners*
Articulate Problem Domain	
Introduce Problem Constraints	
Locate, Select, and Develop Cases	
Construct Case Knowledge Base/ Present to Learners	Articulate Goal(s)/Verify Problem Relate Problem Goals to Problem Domain Clarify Alternative Perspectives Generate Problem Solutions
Provide Knowledge Resources	Gather Evidence to Support/Reject Positions
Support Argument Construction	Determine Validity/Construct Arguments Implement and Monitor Solution Adapt Solution
Assess Problem Solutions	

engaging learners in well-structured and ill-structured problem solving. It is important to note that well-structured and ill-structured problems are not really separate entities, that is, they are not dichotomous. Rather they represent points on a continuum (Reitman, 1965). Where on the continuum any problem resides is a function of the complexity of the problem, clarity of the goal state and the criteria addressing it, the prescriptiveness of the component domain skills, and the number of possible solutions and/or solution paths. The prescriptions provided in this paper represent general recommendations that can be applied, mixed, and matched, depending upon the nature of the problem being solved. Successful implementation of problem-solving instruction requires that designers possess "adequate understanding and training in higher-order problem solving principles and skills such that the necessary expertise can be applied in the process" (McCombs, 1986, p. 78). She concludes that good designers need to be able to "think on their feet." The specific methods that designers choose to use in designing any kind of problem-solving instruction must, as always, rely on the good judgment of the designer.

It is also important to note that the models presented in this paper are not recommended as definitive answers, but rather as works in progress. Experimentation with and assessment of the models along with dialogue among the instructional-design community are needed to validate anything approaching a definitive model for problem-solving instruction. ☐

David H. Jonassen is Professor and head of the Instructional Systems program at the Pennsylvania State University. His publications include the recently completed *Handbook of Research for Educational Communications and Technology*, reviewed in this issue.

Note: Many thanks to Brent Wilson, Marty Tessmer, and the two ETR&D reviewers for their constructive criticism of previous drafts of this paper. I hope that I was able to adequately address their concerns. However, the deeper I got into this paper, the more I realized that it could not adequately exemplify or instruct the many ideas contained in it. That would require a lengthy book. Perhaps in the future after some testing and discussion . . . .

REFERENCES

Anderson, J.R., Farrell, R. & Sauers, R.. (1984). Learning to program in LISP. *Cognitive Science, 8,* 87–129.

Arlin, P.K. (1989). The problem of the problem. In J.D. Sinnott (Ed.), *Everyday problem solving: Theory and applications* (pp. 229–237). New York: Praeger.

Bloom, B.S., Englehart, M.D., Furst, E.J., Hill, W.H., & Krathwohl, D.R. (1956). *Taxonomy of educational objectives: The classification of educational goals: Handbook 1; The cognitive domain.* New York: Longman.

Bodker, S. (1991). Activity theory as a challenge to systems design. In H.E. Nissen, H.K. Klein, & R. Hirschheim (Eds.), *Information systems research:*

Contemporary approaches and emergent traditions. Amsterdam: Elsevier.

Bransford, J. (1994). Who ya gonna call? Thoughts about teaching problem solving. In P. Hallinger, K. Lithwood, & J. Murphy (Eds.), *Cognitive perspectives on educational leadership.* New York: Teacher's College Press.

Bransford, J., & Stein, B.S. (1984). *The IDEAL problem solver: A guide for improving thinking, learning, and creativity. New York: W.H. Freeman.*

Charney, D., Reder, L., & Kusbit, G.W. (1990). Goal setting and procedure selection in acquiring computer skills: A comparison of tutorials, problem solving, and learner exploration. *Cognition & Instruction, 7*(4) 323–342.

Chi, M.T.H., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5,* 121–152.

Chi, M.T.H. & Glaser, R. (1985). Problem solving ability. In R.J. Sternberg (Ed.), *Human abilities: An information processing approach.* New York: W.H. Freeman.

Chi, M.T.H. & Glaser, R., & Rees, E. (1982). Expertise in problem solving. In R.J. Sternberg (Ed.), *Advances in the psychology of human intelligence* (pp. 7–77). Hillsdale, NJ: Lawrence Erlbaum.

Churchman, C.W. (1971). *The design of inquiring systems: Basic concepts of systems and organizations.* New York: Basic Books.

Cognition Technology Group at Vanderbilt (1992). Technology and the design of generative learning environments. In T.M. Duffy & D.H. Jonassen (Eds.), *Constructivism and the technology of instruction: A conversation* (pp. 77–89). Hillsdale, NJ: Lawrence Erlbaum Associates.

Cooper, G., & Sweller, J. (1987). The effects of schema acquisition and rule automation of mathematical problem-solving transfer. *Journal of Educational Psychology, 79,* 347–362.

Covington, M.C., Crutchfield, R.S., Daves, L.B., & Olton, R.M. (1974). *The productive thinking program: A course in learning to think.* Columbus, Ohio: Charles Merrill.

de Jong, T., & Ferguson-Hessler, M.G.M. (1986). Cognitive structures of good an poor novice problem solvers in physics. *Journal of Educational Psychology, 78,* 279–288.

deBono, E. (1974). *CoRT thinking materials.* London: Direct Educational Services.

Dunkle, M.E., Schraw, G., & Bendixen, L.D. (1995, April). *Cognitive processes in well-defined and ill-defined problem solving.* Paper presented at the annual meeting of the American Educational Research Association, San Francisco, CA.

Ernst, G.W., & Newell, A. (1969). *GPS: A case study in generality and problem solving.* New York: Academic Press.

Gagné, R.M. (1966). *The conditions of learning* (1st ed.). New York: Holt, Rinehart and Winston.

Gagné, R.M. (1977). *The conditions of learning* (3rd ed.). New York: Holt, Rinehart and Winston.

Gagné, R.M. (1985). *The conditions of learning* (4th ed.). New York: Holt, Rinehart and Winston.

Gagné, R.M., Briggs, L.J., & Wager, W.W. (1992). *Principles of instructional design* (4th ed.). New York: Harcourt, Brace, & Jovanovich.

Gick, M.L. (1986). Problem-solving strategies. *Educational Psychologist, 21,* 99–120.

Gick, M.L., & Holyoak, K.J. (1980). Analogical problem solving. *Cognitive Psychology, 12,* 306–355.

Gordon, S.E., & Gill, R.T.. (1989). The formation and use of knowledge structures in problem solving domains. Tech. Report AFOSR-88-0063. Washington, DC: Bolling AFB.

Greeno, J. (1978). Natures of problem-solving abilities. In W. Estes (Ed.), *Handbook of learning and cognitive processes* (pp. 239–270). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hayes, J.R. (1981) *The compleat problem solver.* Philadelphia: Franklin Institute Press.

Jacobson, M.J. (1990). *Knowledge acquisition, cognitive flexibility, and the instructional applications of hypertext: A comparison of contrasting designs for computer-enhanced learning environments.* (Doctoral dissertation, University of Illinois).

Jonassen, D.H. (1996). Scaffolding diagnostic reasoning in case-based learning environments. *Journal of Computing in Higher Education.*

Jonassen, D.H., Ambruso, D.R., & Olesen, J. (1992). Designing a hypertext on transfusion medicine using cognitive flexibility theory. *Journal of Educational Hypermedia and Multimedia, 1*(3), 309–322.

Jonassen, D.H., Beissner, K., & Yacci, M.A. (1993). *Structural knowledge.* Hillsdale, NJ: Erlbaum.

Jonassen, D.H., Doricott, D., & Engels, R. (1995). *Comparing the effectiveness of worked examples and trial-and-error learning methods in an engineering microworld.* American Educational Research Association, San Francisco, CA.

Jonassen, D.H., Mann, E., & Ambruso, D.J. (1996). Causal modeling for structuring case-based learning environments. *Intelligent Tutoring Media, 6*(3/4), 103–112.

Jonassen, D.H., & Tessmer, M. (1996). An outcomes-based taxonomy for the design, evaluation, and research of instructional systems. *Training Research Journal.*

Kosonen, P., & Winne, P.H. (1995). Effects of teaching statistical laws on reasoning about everyday problems. *Journal of Educational Psychology, 87,* 33–46.

Kitchner, K.S. (1983). Cognition, metacognition, and epistemic cognition:A three-level model of cognitive processing. *Human Development, 26,* 222–232.

Kitchner, K.S., & King, P.M. (1981). Reflective judgment: Concepts of justification and their relationship to age and education. *Journal of Applied Developmental Psychology, 2,* 89–116.

Kluwe, R.H., & Friedrichsen, G. (1985), Mechanisms of control and regulation and problem solving. In J. Kuhl & J. Beckman (Eds.), *Action control* (pp. 203–217). Berlin: Springer-Verlag.

Kuutti, K. (1996). Activity theory as a potential framework for human-computer interaction research. In B.A Nardi (Ed.), *Context and consciousness: Activity theory and human-computer interaction.* Cambridge, MA: MIT Press.

Lave, J. (1988). *Cognition in practice.* Cambridge, UK: Cambridge University Press.

LeBlanc, S.E., & Fogler, H.S.(1995). *Strategies for creative problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Leont'ev, A.N. (1978). *Activity, consciousness, and personality.* Englewood Cliffs, NJ: Prentice-Hall.

Mayer, R.E. (1989). Models for understanding. *Review of Educational Research, 59*(1), 43–64.

McCombs, B.L. (1986). The instructional systems development model: A review of those factors critical to its implementation. *Educational Communications and Technology Journal, 34,* 67–81.

Meacham, J.A., & Emont, N.C. (1989). The interpersonal basis of everyday problem solving. In J.D. Sinnott (Ed.), *Everyday problem solving: Theory and applications* (pp. 7–23). New York: Praeger.

Merrill, M.D. (1983). Component display theory. In C.M. Reigeluth (Ed.), *Instructional design theories and models.* Hillsdale, NJ: Lawrence Erlbaum.

Nardi, B.A. (1996). *Context and consciousness: Activity theory and human-computer interaction.* Cambridge, MA: MIT Press.

Newell, A. (1980). Reasoning, problem solving and decision processes: The problem space as a fundamental category. In R.S. Nickerson (Ed.), *Attention and performance: Proceedings of the International Symposium on Attention and Performance, VIII.* Hillsdale, NJ: Lawrence Erlbaum.

Newell, A. & Simon, H. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice Hall.

Polson. P., & Jeffries, R. (1985). Instruction in problem solving skills: An analysis of four approaches. In J.W. Segal, S. F. Chipman, & R. Glaser (Eds.), *Thinking and learning skills* (Vol. 1, pp. 417–455). Hillsdale, NJ: Lawrence Erlbaum Associates.

Polya, M. (1957). *How to solve it* (2nd Ed.). New York: Doubleday.

Reed, S. K. (1992). *Cognition: Theory and applications.* Pacific Grove, CA: Brooks/Cole.

Reitman, W. (1965). *Cognition and thought.* New York: Wiley.

Robertson, W.C. (1990). Detection of cognitive structure with protocol data: Predicting performance on physics transfer problems. *Cognitive Science, 14,* 253–280.

Rubinstein, M. (1975). *Patterns of problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Scandura, J.M. (1964). An analysis of exposition and discovery modes of problem solving instruction. *Journal of Experimental Education, 33*(2), 149–159.

Schank, R., & Cleary, C. (1995). *Engines for education.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schön, D.A. (1990). The design process. In V.A. Howard (Ed.), *Varieties of thinking: Essays from Harvard's philosophy of education center* (pp. 110–141). New York: Routledge.

Simon, H.A. (1976). Identifying basic abilities underlying intelligent performance on complex tasks. In L.B. Resnick (Ed.), *The nature of intelligence.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Simon, D.P. (1978). Information processing theory of human problem solving. In D. Estes (Ed.), *Handbook of learning and cognitive process.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Simon, D.P., & Simon, H.A. (1978).Individual differences in solving physics problems. In R. Siegler (Ed.), *Children's thinking: What develops.* Hillsdale, NJ: Lawrence Erlbaum.

Sinnott, J.D. (1989). A model for solution of ill-structured problems: Implications for everyday and abstract problem solving. In J.D. Sinnott (Ed.), *Everyday problem solving: Theory and applications* (pp. 72–99). New York: Praeger.

Smith, P.L., & Ragan, T.J. (1993). *Instructional design.* Columbus, OH: Merrill.

Spiro, R.J., Feltovich, P.J., Jacobson, M.J., & Coulson, R.L. (1992). Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. In T.M. Duffy & D.H. Jonassen (Eds.), *Constructivism and the technology of instruction: A conversation* (pp. 57–76). Hillsdale, NJ: Lawrence Erlbaum Associates.

Spiro, R.J., Vispoel, W., Schmitz, J., Samarapungavan, A., & Boerger, A. (1987). Knowledge acquisition for application: Cognitive flexibility and transfer in complex content domains. In B.C. Britton (Ed.), *Executive control processes.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Spiro, R.J., Coulson, R.L., Feltovich, P.J., & Anderson, D.K. (1988). *Cognitive flexibility theory: Advanced knowledge acquisition in ill-structured domains.* Tech Report No. 441. Champaign, IL: University of Illinois, Center for the Study of Reading.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12,* 257–285.

Sweller, J., & Cooper, G. (1985) The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2,* 59–89.

Tarmizi, R., & Sweller, J. (1988). Guidance during mathematical problem solving. *Journal of Educational Psychology, 80,* 424–436.

Voss, J.F. (1988). Learning and transfer in subject-matter learning: A problem solving model. *International Journal of Educational Research, 11,* 607–622.

Voss, J.F., & Post, T.A. (1989). On the solving of ill-structured problems. In M.T.H. Chi, R. Glaser, &

M.J. Farr (Eds.), *The nature of expertise.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Voss, J.F., Wolfe, C.R., Lawrence, J.A., & Engle, R.A. (1991). From representation to decision: An analysis of problem solving in international relations. In R.J. Sternberg (Ed.), *Complex problem solving.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Ward, M., & Sweller, J. (1990). Structuring effective worked examples. *Cognition and Instruction, 7*(1), 1–39.

Wickelgren, W.A., (1973). *How to solve problems: Elements of a theory of problems and problem solving.* San Francisco, CA: W.H. Freeman.

Wood, P.K. (1983). Inquiring systems and problem structures: Implications for cognitive development. *Human Development, 26,* 249–265.

Appendix

Well-Structured Problem

Domain: Genetics

Problem Statement: Given a monohybrid cross between autosomal traits, determine the genotype of the parents based upon the offspring's phenotype (what the children look like).

List of Prerequisites:

Concepts:

heterozygous
trait
cross
autosome
genetics
recessive
monohybrid
homozygous
autosomal
variable
offspring
characteristic
genes
dominate
phenotype
prodigy
generation
cell
genotype

Rules/Principles:

Mendelian Genetics
PUNNET Square
Division
Ratios

Worked Example:

Problem: Eva the gardener crosses a tall pea plant with a short pea plant. The cross resulted in sixteen peas plants. Eight of the offspring pea plants were tall and eight were short. Tall is the dominant trait. What is the genotype of the parent plants that Eva used to perform the cross?

Note: Height is autosomal.

Step 1: Look at the offspring. There are . . .
Eight tall pea plants
Eight short pea plants

Step 2: Find the ratio of the tall plants to the short plants using your knowledge of ratios and division.

The ratio is 8 to 8 resulting in a 1 to 1 ratio after applying division rules.

Normally at this point we would examine the ratio to determine which trait is dominant (tall or short). Usually, the trait with the more frequencies is dominant. But in this case the ratio is 1:1. So, we would need to look elsewhere to determine dominance.

One way to confirm dominance would be for Eva to cross the offspring and examine the resulting ratio. But in this case, it has been already determined that tall is dominant over short.

Step 3: Choose a variable to represent height. This can be any letter that you prefer. In this case, we will use an "h" to represent height. An uppercase "H" represents the dominate trait (tall) and the lowercase "h" represents the recessive trait (short).

H—dominate trait—in this case Tall

h—recessive trait—in this case Short

Step 4: Determine all genotypes possible

An average cell has two genes which control a specific trait, such as height.

The parents in this cross passed on the two genes (one from each parent) to the offspring. So half of an offspring's genetic make-up came from one parent, and the other half from the other parent.

Determine the possible gene pairs by making as many possible matches with the variables that you decide to use.

Below are the possible pairs for the parents and offspring.

HH Hh hh

Step 5: If a dominate characteristic shows up in a genotype, it masks the recessive characteristic. Meaning, by looking at the offspring plants you will only see tall plants (phenotype). You would not know if plant has a gene for shortness (genotype).

Since tall is dominate in this case there are two possible ways to express the genotype of the tall plant.

HH Hh

Above are the two ways to represent a genotype for dominance in height. We know this because the "H" represents the dominate trait.

Step 6: In the problem, we were told that one of the parent plants was short. Since this is a recessive trait, it ca only be seen (phenotype) when the dominant gene is absent. Since short is the recessive trait, it can only have one possible genotype, "hh."

The genotype of the tall plant is a different story. . . . The tall plant can have two possible combinations (HH or Hh), because it must have at least one dominate gene.

Step 7: To find the genotype of the tall plant, let's turn to the Punnet Square!

a. Along the top of the square we can put the genotype of the short plant with the one gene above each box.

	?	?
?	?	?
?	?	?

Remember the genotype of the short plant is "hh."

	h	h
?	?	?
?	?	?

b. Since, we know that the tall plant has at least one dominate gene ("H"). We can put one "H" on the left side of the Punnet Square by the upper boxes.

	h	h
H	?	?
?	?	?

c. Now lets put in the offspring's genotype. Remember that each parent contributes one gene to the offspring. Therefore, the tall offspring must have received the tall gene from the tall plant since the short plant does not have this gene. The short plant contributed the short genes.

	h	h
H	Hh	Hh
?	?h	?h

d. Now, look at the PUNNET Square. We have two tall plants identified as "Hh." Recall the ratio was 1 to 1. Remember, a short plant's genotype must be "hh." The only way to get the short plants to exist is to have the second trait of the tall plant to be "h." (Since an offspring receives one gene from each parent, both parents must have contributed a gene to form the short plants.)

	h	h
H	Hh	Hh
?	?h	?h

Step 8: In conclusion, the genotype of the tall plant would be "Hh" or heterozygous dominant and the genotype of the short plant would be "hh" or homozygous recessive.

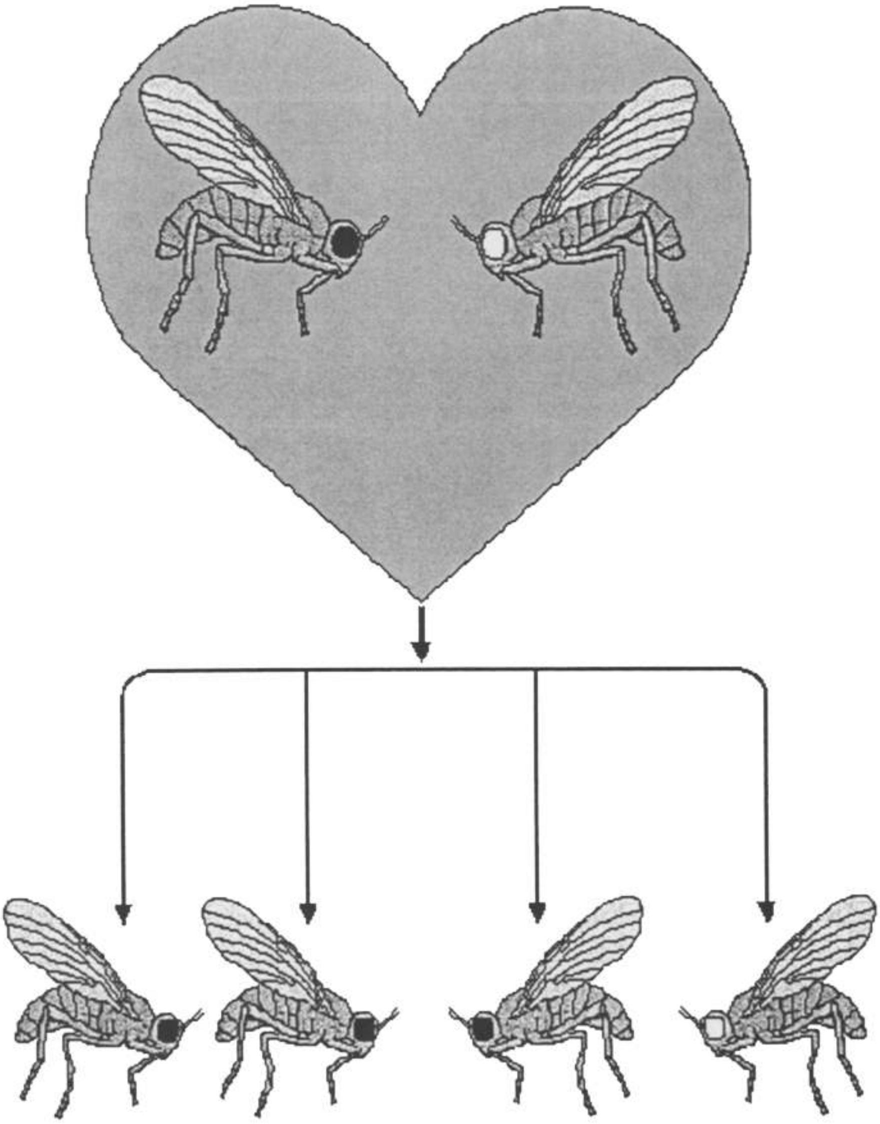
	h	h
H	Hh	Hh
?	hh	hh

Practice Problem:

A man and a women, who both have curly hair, have four children. Three of the children have curly hair and one has straight hair. Curly hair is the dominate trait. What are the genotypes of the parents?

Note: Curly hair is a autosomal trait.

Figure 1a ☐ Conceptual model for problem.



Ill-Structured Problem

This is an ill-structured problem that we have used with instructional design students (thanks to David Birdwell).

Problem Situation: You are contracted by a medical insurance company to help them solve an expensive performance problem. They have been deluged with malpractice suits resulting from gall bladder surgeries. Even though they have won most of the suits, the legal action is expensive and very time-consuming. When physicians diagnose a dysfunctional gall bladder, they are obligated to explain various treatment options to the patient. Additionally, they must explain the procedures entailed by each of those options, along with the inherent risks of each procedure. If the surgeon recommends surgical removal of the gall bladder, then the patient must legally understand the inherent risks. Often, through malfeasance on the part of the surgeon, or the inability of the patient to understand the procedure, a communication gap develops between patient and physician. When complications have arisen during surgery, causing injury or suffering to the patients, many of the patients have sued the physician for malpractice, claiming they were not adequately informed about the risks.

You have been contracted by the insurance company to develop some kind of intervention that will decrease the likelihood of suits being filed when and if any complications arise during surgery.

Problem Domain: The problem domain is surgical practice. More specifically, the problem involves patient understanding of each of the surgical options and acceptance of the inherent risks entailed by that kind of surgery.

Problem Constraints: The important issues here are risk assessment and informed consent. When the spirit or intent of informed consent has been found to be violated, the insurance company employing you has had to pay out large settlements, which drive up the premium costs for malpractice insurance. The company's record is sixty percent victories. However, the court losses have totaled more than $120 million. The insurance company wants you to develop a validated method for instructing patients about inherent surgical risks. They would also like to validate an assessment method that could be found to legally insure that patients understood and accepted the inherent risks of various surgical procedures.

Cases: Roughly 90% of the surgery that is performed treats only a handful of diseases. One of the most common surgical procedures is a cholecystectomy, or surgical removal of the gall bladder when it is diseased. Your task is to develop a validated method for insuring that patients electing to have a cholecystectomy fully understand and legally accept the inherent risks, so that if problems occur, they cannot successfully sue the insurance company.

Additional cases could be presented involving hysterectomies, breast biopsies, and colostomies.

Knowledge Base Made Available to Learners: This includes authentic malpractice court cases with particular attention to criteria related to physician's obligation to inform, samples of patients' medical histories, articles related to the problem of informed consent, samples of legal informed consent forms, and legal definitions of informed consent from Black's Law Dictionary and from the dictionary of the American Medical Association. Videos are also available showing physicians explaining gall bladder disease and its treatment options and possible complications.

Support for Argument Construction: These include templates for legal arguments that attorneys use to prosecute or defend physicians.