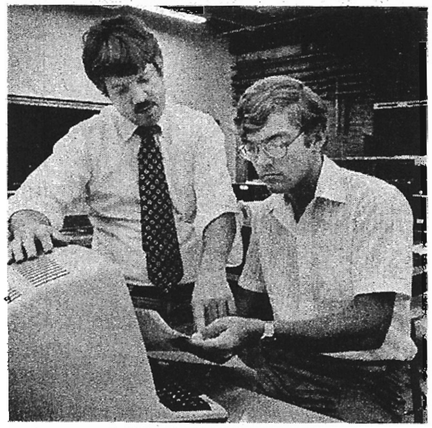

EDUCATIONAL ENGINEERING:

Heuristics for Improving Learning Effectiveness and Efficiency

Karl A. Smith
University of Minnesota

> It is strange that we expect students to learn, yet seldom teach them anything about learning. We expect students to solve problems, yet seldom teach them anything about problem solving. And, similarly, we sometimes require students to remember a considerable body of material, yet seldom teach them the art of memory. It is time we made up for this lack. . . .[1]
>
> —*D.A. Norman*

Educational engineering is based on recent developments in knowledge engineering and cognitive science, two current topics in artificial intelligence. What is the contribution of knowledge engineering (and the broader field of cognitive science) to improving learning efficiency and effectiveness? And what are the implications and importance of these considerations for engineering education?

Peter Drucker coined the terms *efficiency* (doing things right) and *effectiveness* (doing the right thing) in reference to business management.[2] The terms apply as well to learning efficiency (enhancing the rate of learning) and learning effectiveness (enhancing the mastery and retention of facts, concepts, and relationships). Learning effectiveness and efficiency can be enhanced by providing students with strategies that teach them how to learn.

Learning How to Learn

The concept of learning how to learn, first articulated by Bateson,[3] was associated with the then new science of cybernetics. Also called *double-loop learning*, this form of learning involves changes in the governing variables in contrast to *single-loop learning*, which involves learning new strategies to achieve existing governing variables.[4]

The process of learning how to learn is commonly referred to in the cognitive science literature as *metalearning*. The prefix *meta* here means "going beyond," "on a higher level," or "transcendent." In a like manner, *metaknowledge* refers to the structuring of knowledge. Metalearning and metaknowledge are two different but interconnected concepts that characterize human understanding. Learning about the nature and structuring of knowledge helps students understand how they learn and how humans construct new knowledge. Novak and Gowin[5] describe specific strategies for helping students learn about how knowledge is structured and produced. Concept maps (as shown in figure 1) are intended to represent meaningful relationships between concepts in the form of propositions. They are overt, explicit representations of the concepts and propositions a person holds.

In discussing recent improvements in learning efficiency and effectiveness, I will follow the approach of the engineering method, defined by Koen[6] as "the use of heuristics to cause the best change in a poorly understood situation within the available resources." Heuristics from cognitive science that assist in making the best change in students' learning effectiveness and efficiency will be reviewed.

Knowledge Engineering

Some of the contributions of knowledge engineering (and the broader field of cognitive science) to metalearning include models of the learner, expert-novice differences, acquisition of expertise, and knowledge structure and representation.

Models of the Learner. Bruner[13] outlined five models of the learner that serve as a useful guide to the contribution of cognitive science to metalearning:

1) *Tabula rasa* ("one learns from experience") rests on the premise that experience writes on the wax tablet of the mind. According to this model, the order in the mind reflects the order that exists in the world.

2) *Hypothesis generator* learner models react against the passive, tabula rasa models and propose that the learner, rather than being a creature of experience, selects what enters the mind.

3) *Nativism* theories share one central concept: mind is inherently or innately shaped by a set of underlying categories, hypotheses, and organizing experiences. Everything hinges on opportunities to use and exercise the innate powers of mind.

4) In *constructivism* the world is not found, but made, and made according to a set of structural rules that are imposed experiences.

5) The recently developed *novice-to-expert* view begins with the premise that if you want to develop learning strategies, find an expert and examine him or her, then figure out how a novice can become an expert. Computer simulations often are used to identify algorithms and heuristics that will allow a novice to become an expert.

Bruner[14] offered his synthesis of the general models we store in our heads that guide our perception, thought, and talk by saying that "they appear to be diverse, rich, local, extraordinarily generative." Bruner also discussed two modes of thought, each providing distinctive ways of ordering experience, of constructing reality. One mode, the paradigmatic or logico-scientific one, is familiar to engineers and scientists. The other, the narrative mode, deals in human or human-like intention and action, and the vicissitudes and consequences that mark their course. Bruner claims that "great fiction, like great mathematics, requires the transformation of intuitions into expressions in a symbolic system—natural language or some artificialized form of it."[14]

Expert-Novice Differences. Enormous differences of degree and type have been observed in the approach taken by experts, novices, and uninstructed students in solving problems. Novices ask questions such as, "What formula do I know that relates what's given with what I've been asked to find?" They quickly move to a calculation phase and seldom reflect (at least overtly) on what they are doing. Experts ask questions such as, "What are the general principles that apply?" They spend more time thinking about the problem, asking themselves questions, and commenting on their understanding of the problem. Schoenfeld[15] mapped the mathematical problem-solving activities of novices and experts on numerous problems and noted these major differences. Similar results have been found for uninstructed students, novices, and experts solving physics problems.[16]

Modeling the problem-solving methods of experts, and using these models as instructional aids, is an attractive approach, but it is complicated by what experts are able to tell about what they know. Furthermore, although novice-to-expert models are the principal ones followed by cognitive science researchers, there are many other learner models.

Acquisition of Expertise. Expertise appears to be acquired in three stages.[17] In the first phase, cognition or thought, students learn from instruction or observation what knowledge and actions are appropriate. In the associative phase, students practice (with feedback) the relationships discovered or taught in the first phase until they become smooth (fluent and efficient) and accurate (proficient). In the third phase, termed *automaticity*, relationships are

The Role of Heuristics

The word *heuristic* comes from a Greek word meaning "serving to discover." Heuristics have become very popular in the cognitive science literature, and interest is growing once again in the problem-solving[7,8] and engineering-method[9] literature. The work of one of the most prolific promoters of heuristics, George Polya,[9,10] is aimed explicitly at teaching the young how to be better problem solvers.[11] *Heuristic* is currently used as an adjective in the sense of "guiding discovery" or "improving problem solving."

Although difficult to define, heuristics are relatively easy to identify using the characteristics listed by Koen.[9] Their use does not guarantee a solution; two heuristics may contradict or give different answers to the same question and still be helpful. Heuristics help solve unsolvable problems or reduce the time needed to find a satisfactory solution. The heuristic depends on the immediate context instead of absolute truth as a standard of validity.

Learning to use heuristic strategies is necessary but not sufficient to ensure competent problem-solving performance. Schoenfeld asserts that equivocal results have occurred because the complexity of heuristic strategies, and the amount of knowledge needed to implement them, have been underestimated.[12] If the teaching of heuristics is to be effective, it must focus not only on the heuristics themselves but on when and where to apply them.

"compiled" continuously until they do not require much thought on the learner's part.

Dreyfus and Dreyfus[18,19] extended and elaborated on these phases of skill acquisition and proposed five stages—novice, advanced beginner, competent performer, proficient performer, and expert. The novice knows basic facts about a subject and context-independent rules for using those facts. The advanced beginner can use examples to formulate rules for action, taking context into account. The competent performer is personally involved, goal-oriented, and able to reason analytically and act without conscious thought about the rules. The proficient performer can recall whole situations and apply them without having to break them down into smaller components. The expert makes little conscious use of analytical reasoning, has little awareness of the skill, is fully involved in the situation, and seems to operate by visualizing and manipulating whole objects and situations.

Performance of tasks at the expert level is usually smooth and proficient; however, experts are generally not consciously aware of the processes they use. Michael Polanyi[20] refers to this type of knowledge as "tacit." Paul Johnson [17] has termed this fact the "paradox of expertise"—i.e., the very knowledge we wish to represent in a computer program, as well as the knowledge we wish to teach others, often turns out to be the knowledge we are least able to talk about.

Knowledge Structure and Representation. Research in cognitive science has contributed to our understanding of knowledge structure, representation, and construction. Researchers in artificial intelligence have experimented with knowledge representation systems and inference procedures. The modeling of learning, however, turned out to be much more difficult than anyone in the AI community expected. For example, sequential readiness assumptions may hold for some simple tasks and for young children; however, adolescent and adult structures of knowledge and individual differences are often uneven and nonlinear.[21]

Constructing Knowledge Bases

Outlines and notes are the main methods that students use to organize knowledge externally. These approaches appear to serve their purposes since the majority of student learning involves rote memorization. Meaningful learning, on the other hand, requires more powerful representation strategies. A concept map (a type of spatial learning strategy) summarizing numerous forms of knowledge representation is shown in figure 1. Concept maps require representation of relationships between concepts; they facilitate abstraction and deep processing. Unlike more content-dependent techniques (matrixing, flowcharting, constructing pictures or graphs, for example) these systems can be used in a wide variety of contexts. Concept maps are intended to represent meaningful relationships between concepts in the form of propositions.[5] In its simplest form, a concept map would be just two concepts connected by a linking word to form a proposition. Unlike outlines, concept maps show key concepts and propositions explicitly and concisely, visually emphasizing both hierarchical relationships between concepts and propositions, and cross links between sets of concepts and propositions.

How the process of constructing knowledge bases (or representations) assists students in learning how to learn will be described later. Additional strategies for representing knowledge have been described.[22]

Expert Systems— A Student Project

At the University of Minnesota we involved our students in building small expert systems by requiring them to construct explicit knowledge representations. Our primary purpose was to familiarize them with this approach in a course on the application of operations research techniques in engineering. There were several unanticipated side effects. Not only were students much more enthusiastic than we had expected, they mastered content that we had not expected them to. They formulated rules for design and decision making that showed they not only had reviewed a great deal of information, but had reviewed it selectively and purposefully. The outcome of this procedure is described briefly below and in more detail elsewhere.[23]

A small expert-system shell[24] has been an indispensable part of the

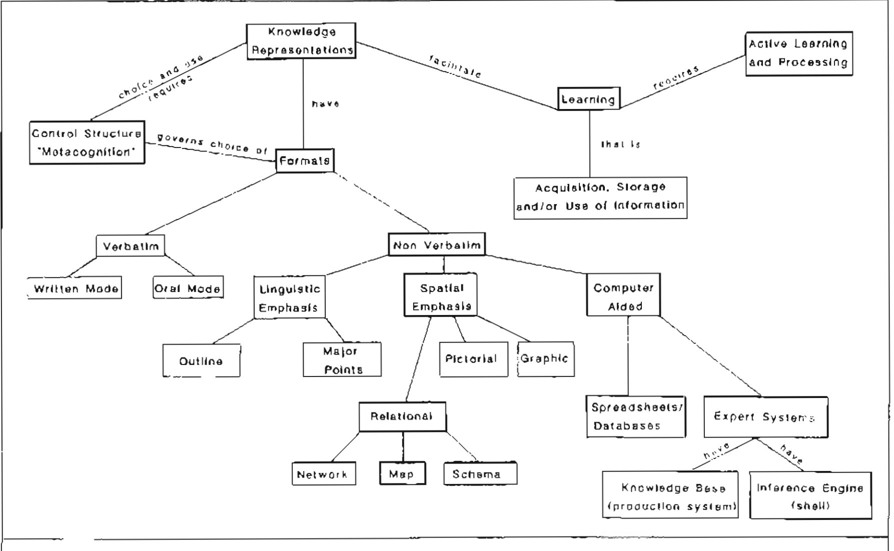

Figure 1. Concept map for constructing knowledge bases.

way we have used knowledge bases as a teaching tool. The shell, written in Pascal, runs on a personal computer with 128K memory. The knowledge bases constructed by the students are stored in text files, according to a simple and flexible format. These files are input data for the shell, which will read, parse, check, and interpret the text. The shell then permits users to interact with the knowledge base.

The knowledge base itself is divided into three parts: 1) a set of numbered decisions, 2) a series of questions that solicit the information necessary to select an appropriate decision (associated with each question is a limited number of answers), and finally, 3) a set of production rules. Each rule has the formation

IF <condition> THEN <decision>

where the condition is a Boolean expression relating to answers to the questions and perhaps also to decisions.

In a typical project students built a knowledge base to select an urban transportion system. The decisions in that case consisted of a list of transportation alternatives, such as

Decision 1 buses

Decision 2 light rail

The questions had to do with the range of lane capacity needed, the maximum possible investment, the speed required, the levels of acceptable noise and pollution, and so on. A typical question might be

Question 3: "What is the range of lane capacity you need?"
Answer 1: "Between 15,000 and 20,000 spaces per hour."
Answer 2: "Between 8,000 and 15,000."

An example of a production rule might be

IF (Q2 Ans3) and (not Dec4) and (Q5 Ans1 or Q6 Ans2) THEN Dec2.

In the interaction between a user and the expert system shell, the shell will ask questions, record and remember the user's responses, and systematically test the rules until it finds a valid one. It will then print out the appropriate decision. An important feature is that the user can ask the shell "why?" at any stage. The knowledge base may contain reasons (text strings) associated with each question and each rule. These should explain briefly the reasoning behind the questions and rules respectively. When users ask "why?" they are first given the reason associated with the current question. If they ask "why?" again they are given the reason associated with the rule that the shell is currently testing.

We have found the following sequence to be effective in the classroom:

1) We first introduce the concept of an expert system and knowledge base, explain the structure of the production rules, and demonstrate a small system on the computer.

2) We divide the class into groups of two's or three's and ask them to suggest topics that would lend themselves to this kind of approach. The ensuing discussion highlights the differences between suitable topics and unsuitable ones.

3) Each group then is required to construct a knowledge base as a homework assignment over a period of one or two weeks. Students are told to pay particular care to the explanation facility and are required to implement and demonstrate their work, using the shell. This allows faculty and fellow students a chance to critique the assignments.

We have found that students adapt quickly to this formal structure, learn to exploit it, and get a solid sense of achievement when they implement their work.

Databases and Spreadsheets. The expert system described above is an example of "build/run" software. It enables students to construct a knowledge base, operate on it, and examine the results. Standard database and spreadsheet programs facilitate the same operations. Students can compare, merge, and test information in a database; a spreadsheet allows them to introduce and simulate mathematical relationships. A successful spreadsheet application must be constructed carefully of knowledge and rules (i.e., values and formulas) so that one might later observe the rippling effect of column by column output of the calculated values. Similarly, a database must be constructed before various sorting and merging functions can be used. These programs provide a means to represent and manipulate knowledge.

Students adapt quickly to this formal structure, learn to exploit it, and get a solid sense of achievement.

Instructional Use of Knowledge Representations

The strategies described above are techniques for externalizing concepts and propositions. However, learning the meaning of a piece of knowledge requires dialogue, exchange, sharing, and sometimes compromise. When spatial learning strategies are used in groups of two or three students, such strategies can serve a useful social function and lead to lively classroom discussion. Cooperative learning groups,[25-27] an active learning technique, has shown similar positive outcomes. Preparing to teach or tutor another, whether or not any teaching is actually done, results in greater achievement and appreciation of the subject and other class members.[28,29] Cognitive psychology researchers have shown that talking with peers and preparing to teach are the two principal contributions to the development of expertise.[30]

The learner must actively analyze the structure in order to construct a spatial representation. In addition to training and encouraging students to construct knowledge bases to assist in learning, instructors could incorporate a variety of forms of knowlege bases in their lectures or handout materials. Lecture materials can be modified to incorporate some of these ideas.[31] The construction of knowledge bases can be incorporated in evaluation procedures with, for example, a scoring procedure developed for concept mapping. We have required students to build the knowledge representation for expert systems in exams.

Many recognize microcomputers as a revolution in education not to supplant traditional educational processes, but rather to supplement them by allowing students to experiment with many different situations and to "instruct" the machine rather than be "instructed."[32,33] Programs that enable one to modify data and quickly recalculate are excellent tools for sensitivity analysis and encourage a deeper understanding of the behavior of the system being studied. Once a knowledge representation is constructed, one is free to operate on it with the tools of a particular package using a "what if" approach.

Selection of Knowledge-Representation Strategies

Metalearning requires a capability for examining one's own knowledge and thoughts and then modifying them accordingly. The "control structure" that can accomplish this modification is called *metacognition* in the cognitive science literature. Flavell's generally accepted definition of metacognition is as follows:

> Metacognition refers to one's knowledge concerning one's own cognitive processes and products or anything related to them, e.g., the learning-relevant properties of information or data. Metacognition refers, among other things, to the active monitoring and consequent regulation and orchestration of these processes in relation to the cognitive objects on which they bear, usually in the service of some concrete goal or objective.[34]

Metacognition has two separate but related aspects: 1) knowledge and beliefs about cognitive phenomena, and 2) the regulation and control of cognitive actions.[34] Since several strategies are available to each learner, the student should be helped to choose appropriate strategies for each learning task. Bruner argues against promoting only one model of the learner and suggests instead that the best approach is a reflective one that allows one to "go meta."[35]

Problem formulation is neglected in much of engineering education. It requires the capability to learn how to learn.

Implications for Engineering Education

In real-world engineering practice, problems do not present themselves to the practitioner as givens. Problem formulation—the process by which we define the decisions to be made, the ends to be achieved, and the alternative means that may be chosen—is neglected in much of engineering education. Problem formulation requires the capability to learn how to learn and to reflect-in-action. According to Schon reflection-in-action exemplifies professional activity:

> When someone reflects-in-action, he becomes a researcher in the practice context. . . . He does not separate thinking from doing, ratiocinating his way to a decision which he must later convert to action. Because his experimenting is a kind of action, implementation is built into his inquiry.[4]

In *Educating the Reflective Practitioner* Schon describes his approach to the development of professional practice skills. He writes:

> Designing, both in its narrower architectural sense and in the broader sense in which all professional practice is designlike, must be learned by doing. . . . A designlike practice is learnable but is not teachable by classroom methods. And when students are helped to learn design, the interventions most useful to them are more like coaching than teaching—as in a reflective practicum.[36]

Meaningful learning and especially learning how to learn is enhanced by talking with peers and preparing to teach others. A key to learning how to learn, therefore, is to get students involved in the construction of knowledge representations. Schoenfeld acts as a roving consultant while the class breaks into small groups to work on mathematics problems. He has found that asking the following three questions promote the development of metacognitive skills.[37]

1) What (exactly) are you doing? (Can you describe it precisely?)
2) Why are you doing it? (How does it fit into the solution?)
3) How does it help you? (What will you do with the outcome when you obtain it?)

Conclusion

The development of higher cognitive skills that enable students to be independent learners and independent, creative problem-solving users of their knowledge is a very important goal for educators. Providing students with an active learning environment where they can get involved with the material to be learned in a mutually supportive situation with other people, and providing them with tools such as the ones described here, will contribute to meaningful learning.

Even if learning, thinking, and problem-solving strategies, whether general or specific, are shown to exist, it might not be possible to teach them directly. Perhaps they must emerge spontaneously as a result of experience. The current conception is that metacognition—conscious awareness and control of cognitive processes—emerges only as knowledge and skills in a particular do-

Meaningful learning is enhanced by talking with peers and preparing to teach others.

main become quite well developed. At the very least, it should be possible to select and design experiences to result in a more rapid and complete emergence of such skills. A key to the success of developing students' skill at using these strategies is for faculty to incorporate them in their handouts, exercises, lectures, assignments, and exams.

References

1. Norman, D. A., "Cognitive Engineering and Education," in D. T. Tuma and F. Reif (eds.), *Problem Solving and Education: Issues in Teaching and Research*, Erlbaum, 1980, pp. 97-107.

2. Drucker, P. F., *The Effective Executive*, Harper & Row, 1964.

3. Bateson, G., "Social Planning and the Concept of Deuterolearning" (1942), reprinted in *Steps to an Ecology of Mind*, Chandler, 1972.

4. Schon, D. A., *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, 1983.

5. Novak, J. D. and D. B. Gowin, *Learning How to Learn*, Cambridge University Press, 1984.

6. Koen, B. V., *Definition of the Engineering Method*, ASEE, 1985, page 70.

7. Rubinstein, M., *Tools for Thinking and Problem Solving*, Prentice-Hall, 1986.

8. Nickerson, R. S., D. N. Perkins, and E. E. Smith, *The Teaching of Thinking*, Erlbaum, 1985.

9. Koen, B. V., "Toward a Definition of the Engineering Method," *Engineering Education*, vol.75, no.3, Dec. 1984, pp. 150-155. (See also ref. 6.)

10. Polya, G., *How to Solve It*, Princeton University Press, 1945.

11. Newell, A., "The Heuristic of George Polya and its Relation to Artificial Intelligence," in R. Groner, M. Groner, and W. F. Bischof (eds.), *Methods of Heuristics*, Erlbaum, 1983, pp. 195-244.

12. Schoenfeld, A. H. "Artificial Intelligence and Mathematics Education: A Discussion of Rissland's Paper," in E. A. Silver (ed.), *Teaching and Learning Mathematical Problem Solving*, Erlbaum, 1985, pp. 177-187.

13. Bruner, J., "Models of the learner," *Educational Researcher*, June/July 1985, pp. 5-8.

14. Bruner, J., *Actual Minds, Possible Worlds*, Harvard University Press, 1986, p. 15.

15. Schoenfeld, A. H., *Mathematical Problem Solving*, Academic Press, 1985.

16. Champagne, A. B., L. E. Klopfer, and R. E. Gunstone, "Cognitive Research and the Design of Science Instruction," *Educational Psychologist*, vol. 17, no. 1, 1982, pp. 31-53.

17. Johnson, P. E., "What Kind of Expert Should a System Be?" *The Journal of Medicine and Philosophy*, vol. 8, 1983, pp. 77-97.

18. Dreyfus, H. L. and S. E. Dreyfus, *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*, The Free Press, 1986.

19. Dreyfus, H. and S. Dreyfus, "Why People May Never Think Like Computers," *Technology Review*, vol. 89, no. 1, 1986, pp. 44-61.

20. Polanyi, M., *The Tacit Dimension*, Doubleday, 1973.

21. Foster, S. F., "Ten Principles of Learning Revised in Accordance with Cognitive Psychology: With Implications for Teaching," *Educational Psychologist*, vol. 21, no. 3, 1986, pp. 231-243.

22. Smith, K. A., A. M. Starfield, and R. Macneal, "Constructing Knowledge Bases," in L.P. Grayson and J.M. Biedenbach (eds.), *Proceedings, Fifteenth Annual Frontiers in Education Conference*, IEEE/ASEE, 1985, pp. 374-382.

23. Starfield, A. M., *et al.*, "Mastering Engineering Concepts by Building an Expert System," *Engineering Education*, vol. 74, no. 2, Nov. 1983, pp. 104-107.

24. Starfield, A. M., S. Adams, and A. Blelock, "A Small Expert System Shell and its Applications." *Proceedings, Fourth Annual Phoenix Conference,* IEEE, 1985, pp. 262-267.

25. Smith, K. A., "Cooperative Learning Groups," in S. Schomberg (ed.) *Strategies for Active Learning and Learning in University Classrooms*, University of Minnesota, 1985, pp. 18-26.

26. Smith, K. A., D. W. Johnson, and R. T. Johnson, "Structuring Learning Goals to Meet the Goals of Engineering Education," *Engineering Education*, vol. 72, no. 3, 1981, pp. 221-226.

27. Smith, K. A., A. Wassyng, and A. M. Starfield, "Development of a Systematic Problem Solving Course: An Alternative to the Use of Case Studies," in L. P. Grayson and J. M. Biedenbach (eds.), *Proceedings, Thirteenth Annual Frontiers in Education Conference*, IEEE/ASEE, 1983, pp. 42-46.

28. Goodlad, S., *Learning by Teaching: An Introduction to Tutoring*, Community Service Volunteers, 1979.

29. Annis, L. F., "The Process and Effects of Peer Tutoring," *Human Learning*, vol. 2, no. X, 1983, pp. 39-47.

30. Champagne, A. B., L. E. Klopfer, and R. F. Gunstone, "Cognitive Research and the Design of Science Instruction," *Educational Psychologist*, vol. 17, no. 1, 1982, pp. 31-35.

31. Day, R. S., "Teaching from Notes: Some Cognitive Consequences," in W. F. McKeachie (ed.), "Learning, Cognition, and College Teaching," *New Directions for Teaching and Learning*, no. 2, 1980, pp. 95-112.

32. Wassyng, A. and S. Sharp, "Effective Teaching in Engineering Using Computer Aided Learning," *Proceedings of the North Midwest Section Meeting*, ASEE, 1985, pp. 49-60.

33. Wassyng, A., K. A. Smith, and S. Sharp, "Personal Computers in Education," in S. Schomberg (ed.), *Strategies for active teaching and learning in university classrooms*, University of Minnesota, 1985, pp. 18-26.

34. Flavell, J. H., "Metacognitive Aspects of Problem Solving," in L. Resnick (ed.), *The Nature of Intelligence*, Erlbaum, 1976, pp. 232.

35. Garafolo, J. and F. K. Lester, "Metacognition, Cognitive Monitoring and Mathematical Performance," *Journal for Research in Mathematics Education*, vol. 16, 1985, pp. 163-176.

Schon, D. A., *Educating the Reflecting Practitioner*, Jossey-Bass, 1987, p. 157.

37. Schoenfield, A. H., "Metacognitive and Epistemological Issues in Mathematical Understanding," in E. A. Silver (ed.), *Teaching and Learning Mathematical Problem Solving,* Erlbaum, 1985, pp. 361-379.

Karl A. Smith is an associate professor of metallurgical engineering at the University of Minnesota. He is a frequent writer and speaker on the use of active learning strategies, including cooperative group learning and structured controversy discussion, knowledge representation and expert systems, and the use of personal computers as instructional tools.