SCHEMA-BASED PROBLEM-SOLVING LABS

Roman Taraban
and
Brian MacWhinney
Carnegie-Mellon University

In: Proceedings of the 1985 IBM Academic Information Systems University AEP Conference. Alexandria, VA, pp. 407-411.

In order to be an expert at something, one has to know a lot about it. An expert problem-solver simply knows more than a novice. We might think of the expert's knowledge as the *contents* of a textbook or manual. Just as important as the contents is the *index* to this material—you need to know under what conditions to apply the knowledge that you have. Larkin (Larkin, McDermott, Simon & Simon, 1980) has pointed out that typical texts on kinematics, for example, present and explain the basic kinematic equations. However, few focus on the process of selecting the appropriate equation(s) in problem-solving situations. This observation that knowing the conditions for applying knowledge may be as important as the knowledge itself is one outcome of our recent use of computer-based labs in an introductory psychology course at Carnegie-Mellon University. After describing the current organization and use of these labs, we will present a model for further development. This schema-based model organizes knowledge of problem-solving methods in a decision network that is used to access that knowledge. Our primary purpose in using this model is to develop a uniform problem-solving environment in which the student can successfully tackle a diverse range of typical real-life problems.

Overview of the labs. Twelve labs were used by about 100 students this past spring as a regular part of "Cognitive Processes"--a course that is part of the core curriculum in the College of Humanities and Social Sciences. We have pursued two general objectives for the course and for the labs. One is to give students insight into their own cognitive processes. These include perception, memory, and problem-solving strategies. The second is to help them improve their problem-solving skills. We want the students not only to learn about problem-solving, but also to begin applying what they learn to their studies in school and to decision-making in and outside of school. In a number of the labs, we have been able to provide students with some measure of control over their learning processes. For example, in some of the labs students practice particular mnemonic skills such as the method of loci and the keyword method at their own pace and level. The labs also provide examples of machine learning and intelligence. In one lab students observe the computer as it solves a puzzle (Missionaries and Cannibals) using a means-ends strategy; and in another lab the computer generates an EPAM discrimination net for paired associates.

The labs fall into three general areas: problem-solving, the mechanics of processing, and cognitive skill development. In the problem-solving labs students solve puzzles. The student creates a record of puzzle-solving moves and then analyzes these as problem-solving strategies. The mechanics of processing labs are mini experiments in which the students are able to recreate some of the basic psychological findings in perception and memory. In these they run themselves a subjects and then analyze their own data, in addition to the data of one or two lab partners. The skills labs give students practice with specific techniques that improve their ability to memorize. It should be mentioned that the formats of several of the labs are based on programs BASIC and distributed by COMPress (Bewley, 1979) and

CONDUIT (Fischler, Griggs, Warner, Sherman & Levy, 1979; Keenan, 1982).

In general our labs consist of the following components: an introduction intended to motivate the lab and to provide instructions for running it, the lab itself, the theory underlying the lab, discussion questions for a lab report, sample output and calculations, and in some cases, computer simulations of learning or problem-solving behavior. These components are implemented as options in the menu for the program. The student, for example, is able to go back and review the instructions, view sample calculations, or read the theory underlying the lab. The labs are available on master diskettes. Lab output and discussion questions are written to a diskette provided by the student.

Students ran these labs in one of the public IBM/PC clusters at CMU. These rooms were reserved for weekly lab sessions. There were 15 to 20 students in a section and each session lasted 50 minutes. There were two teaching assistants present to get students started and to answer questions, although the labs were designed to minimize the need for outside help. The labs were written in the C86 programming language. They were implemented using the Unix C window-management routines called CURSES, which were modified to operate under IBM DOS. The labs were able to write the output file to the student's diskette on either the IBM/PC or XT. And finally, the output file could be printed in a networked configuration using laser printers in the PC cluster.

Specific examples of labs. To give you an idea of what these labs are like we will briefly discuss examples of each type. A prototypical problem-solving lab uses the Tower of Hanoi puzzle. In general, from one to nine donut-shaped disks of different diameters are stacked in pyramid fashion on a source peg. The student must move these disks to the goal peg under the following move constraints: only one disk can be moved at a time, a disk cannot be slipped out from under another disk in order to move it, and a larger disk cannot be placed on top of a smaller disk. A record of problem moves is written to the student's output diskette, along with the minimum number of moves required for the problem and the number of moves actually made. At the end of the lab, students copy the discussion questions for the lab to their output diskette. These questions are the basis for a written report that is evaluated by teaching assistants for the course. The questions guide students through an examination of the general method and specific strategies that they adopted for solving the problems. Students evaluate their own strategies for effectiveness and are encouraged to consider ways of generalizing and improving these strategies. In most cases students exchange data with a partner and use their own and their partner's data in discussing task performance.

The labs in the mechanisms of processing category are primarily mini-experiments in the areas of perception and memory. A typical experiment in this group uses the Muller-Lyer perceptual illusion. The student is shown two lines. The bottom line has arrows attached on the end that are pointing inward and the top line has arrows pointing outward. The student must judge if the top line is the same length as the bottom line, or if it is longer or shorter. A robust effect in this experiment is that the top line must be longer than the bottom line for the student to judge the two lines as being equal. This nicely points out the fact that perception is not exclusively the result of objective stimulus factors, but that cognitive expectations and biases actually affect what we perceive. Another typical experiment in this group uses the "Stern-berg task." In this experiment students hold a list of digits in short-term memory, then scan this list for the presence of a probe digit. Output from this experiment allows students to estimate the rate for scanning items in short-term memory, which, incidentally, is about 50 msec per item, a very fast process. These labs are intended to show students that some basic aspects of cognition are unexpected, but that many are, nevertheless, quantifiable and predictable.

SOCIAL SCIENCES

Let us consider for a moment the practical nature of such a problem-solving network. If, for example, a student is faced with the problem of finding an apartment, he might first categorize the problem as one of decision-making. He would then identify the type of decision-making Problem that he faces based on the task itself and on his goals. Having identified a method, like *satisficing*, he could follow a sequence of operations to reach a solution. If a student must prepare for a chemistry test, he might identify this as a memory problem, choose the best method for the particular material—*e.g*. the method of loci—and apply it.

There are clearly some attractive features in a schema-based organizational framework for planning and developing computer instruction. It extends the traditional notion of learning objectives (Potts, 1985) to include multiple objectives within a unified structure This could be important. Instruction often focuses on discrete bits of knowledge (at least from the learner's point-of-view) and although we may expect students to integrate knowledge, they may not always do so. In fact, the ability to integrate and use a large and diverse knowledge base distinguishes expert problem solvers from novice problem solvers (Chi, Feltovich & Glaser, 1981; Larkin, 1983; Larkin, McDermott, Simon & Simon 1980). As already suggested, the structure in a problem-solving schema allows one to more readily incorporate an instructional tutor. This is because the schema is traversed in stages. At each point it should be possible to predict the most likely errors or misconceptions and to provide for their remediation. Finally, there is no apparent reason why this same framework could not be used as a powerful and effective tool for software development in areas outside of psychology.

REFERENCES

Bewley, W. L. (1979). *Cognitive Psychology: A Computer Oriented Laboratory Manual.* Wentworth, NH: COMPress.

Chi, M., Feltovich, P.J. & Glaser, R. (1981). Categorization and Representation of Physics Problems by Experts and Novices. *Cognitive Science, 5,* 121-152.

Fischler, I. S., Griggs, R. A., Warner, S. A., Sherman, M. E., Levy, C. M. (1979). *Laboratory in Cognition and Perception.* Iowa City, IA: CONDUIT.

Hayes, J. R. (1981). *The Complete Problem Solver.* Philadelphia: The Franklin Institute Press.

Keenan, J. M. (1982). *Computer Lab in Memory and Cognition.* Iowa City, IA: CONDUIT.

Larkin, J. (1983). The Role of Problem Representation in Physics. In D. Gentner & A. L. Stevens (Eds.), *Mental Models.* Hillsdale, NJ: Erlbaum.

Larkin, J., McDermott, J., Simon, D. & Simon, H. (1980). Expert and Novice Performance in Solving Physics Problems. *Science, 208,* 1335-1342.

Potts, J. F. (1985). Ingredients For Quality Education Software. Presentation at the 1985 IBM AEP Conference in Alexandria, VA.

FIGURE 1. A portion of a problem-solving schema for decision-making.