Computers and Problem Solving: A Workshop for Educators

David Moursund

About the Author

David Moursund has been teaching and writing in the field of computers in education for the past 20 years. He is a professor at the University of Oregon in the Department of Curriculum and Instruction. He has run master's and doctorate programs in computer education since the early 1970's.

His accomplishments in computer education include:

- Author or co-author of 18 books and numerous articles.
- Chairman of the Association for Computing Machinery's Elementary and Secondary Schools Subcommittee, 1978-82.
- President of the International Council for Computers in Education, 1979-84.
- Chief Executive Officer of the International Council for Computers in Education and Editor-in-Chief of The Computing Teacher.

International Council for Computers in Education

This book is published by the International Council for Computers in Education, a nonprofit, tax-exempt professional organization. ICCE is dedicated to improving educational uses of computers and to helping both students and teachers become more computer literate. ICCE publishes The Computing Teacher, a journal for teachers of teachers. It also publishes a number of booklets of interest to educators. Write for a free catalog at the address given below.

The booklet prices given below are for prepaid orders. On other orders a $2.50 invoicing fee will be added. Please add shipping charges of $2.50 for orders up to $20; $3.50 for orders between $21 and $50 ; and 5% for orders above $50.

Quantity	Price (U.S.)
1-4 copies	$8.00 ea.
5-9 copies	$7.20 ea.
10-99 copies	$6.40 ea.
100+ copies	$5.60 ea.

Please place your order with:

International Council for Computers In Education

University of Oregon

1787 Agate St.

Eugene, OR 97403-9905

Ph. 503/686-4414

Computers and Problem Solving:
A Workshop for Educators

Contents

ISBN 0-924667-34-6

Original Preface

Over the past few years I have presented a large number of Leadership Development Workshops for educators involved with instructional uses of computers. Many of these workshops contain a major component on roles of computers in problem solving.

Gradually the problem-solving component of these workshops has taken on a life of its own and has grown into a self-contained workshop. Typically this workshop is a half-day or a full day in length, although it expands or contracts to fit the particular amount of time available. The materials are easily expanded to a much longer workshop, since problem solving is a relatively large and complex field.

Problem solving is an important aspect of every academic discipline, and computers are useful aids in solving a wide variety of problems. Thus, my problem solving workshops are designed for mixed audiences. They typically include a mixture of elementary and secondary school teachers and administrators, as well as computer coordinators and college faculty. Moreover, the computer backgrounds and interests of workshop participants vary widely.

Needless to say, preparing and presenting a workshop to meet the needs of such a diverse group is a challenging task. After a workshop is completed, I mentally review the content and process of the workshop. I search for strengths and weaknesses. What went well? What needs improvement?

One conclusion I have reached is that workshop participants need to have in hand and carry away a written document that captures the essence of the content and process of the workshop. The document needs to be relatively easy and fun to read. It needs to contain some new ideas and to reinforce ideas covered in the workshop. It needs to suggest applications of the workshop content and to encourage participants to use some of these applications. In a nutshell, that describes the purpose of this booklet.

For me, a workshop is a delightful environment for interacting with educators, trying out new ideas, and working to improve our educational system. A workshop is a balance between content and process. It involves substantial interaction among the participants and with the workshop facilitator.

It is relatively easy to capture the content of a workshop in print. But print does not lend itself well to capturing process. Moreover, reading a book all by yourself is quite a bit different than participating with an excited group of educators in a group learning process. Thus, readers of this booklet will have to mentally recreate the excitement and the group process by drawing upon their own teaching and workshop experiences.

I want to thank all people who have participated in my workshops. They have allowed me to grow, and they have contributed many of the ideas in this booklet.

Dave Moursund

May 1986, 1988

Preface for the 2004 Reprinting

I am older than I used to be. And, I am younger than I eventually will be. It is interesting to look forward, to see what is coming down the pike. It is also interesting to look back, and to see what was.

During the past few days, I participated in a Blue Ribbon Panel hosted by Learning Point Associates. The current North Central Regional Educational Laboratory (NCREL) is a wholly owned subsidiary of Learning Point Associates. The purpose of the meeting of the Blue Ribbon Panel was to look toward the future of Information and Communication Technology (ICT) in Education and to aid NCREL in its work. During the meeting I had the opportunity to listen to a large number of very bright people share their insights into the future of ICT in education. In preparation for the meeting, the participants were provided with some documents that I have written about possible futures. In addition, I participated in the discussions at the meeting.

During my "off" time at the meeting, I reformatted *Computers and Problem Solving: A Workshop for Educators* into the form you are now reading. That allowed me to carefully read the old document. I found it interesting that many of the old ideas are still quite current.

How can that be? The answer lies in the nature of problem solving. While computer technology continues to make rapid strides, the underlying ideas of problem solving with and without computer technology remain much the same. In the future, as in the past, problem solving lies at the very heart of every academic discipline. In the future, as in the past, higher-order thinking and problem solving are core goals in education.

In my opinion, *Computers and Problem Solving: A Workshop for Educators* is still a very useful book. The original text has been modified by the addition of a few commas and a change of the word "which" to "that" in a couple of places. The original illustrations (designed to lighten up the text) have not been included. Appendix B, which was written for use in a revision of the book that did not occur, has been added for historical purposes. I am pleased that this book can be made available (at no charge) to those who wish to access it through the Web.

David Moursund

November 2004

Part 1: What is a Problem?

Getting Started

Note to readers: This booklet is written in the first person, and it is somewhat like a transcript of a workshop session. This is to help capture some of the flavor of a workshop. Actually, this booklet is a composite of many workshops, with additional information to increase its usefulness.

This is a workshop on roles of computers in problem solving and possible effects these roles will have on education. This workshop—the physical facility, the participants and the facilitator—constitutes a learning environment. Each of you is an important part of the environment. Sharing ideas among yourselves will be an important workshop activity.

The workshop has been designed to help you gain increased knowledge about roles of computers in problem solving. Take a minute to review in your mind some of the things you know about problem solving. What are some of the most important ideas? When you think about being asked to solve a problem, what do you feel? Do you consider yourself to be good at solving problems? Are you good at helping other people learn to solve problems? What do you expect to happen during the next couple of hours here in this workshop? How might this workshop help you?

One can view this workshop as an exploration of the problem of determining and handling the problem of roles of computers in problem solving in education. [Note to reader: When presented orally to workshop participants, the last sentence is likely to befuddle the mind. It's sort of like the idea of thinking about thinking, or thinking about thinking about thinking. This befuddlement is intended. It can help break a mindset of preconceived notions about the possible content of the workshop and/or about problem solving.]

Exercise. Let's begin the workshop interaction with an exercise. All of you are well-educated, intelligent, functional educators. In your everyday lives, at home, at work, and at play, you encounter a variety of problems. You cope with or solve these problems—you do what needs to be done.

Right now I want you to think back to some problem you have recently encountered and solved. Get the problem firmly in mind. What were the circumstances in which you encountered the problem? Were there other people involved, or were you alone? What did you see, hear, and feel as you encountered the problem?

Recreate the problem solving process that you went through as you solved the problem. What did you do first? What did you do next? Did you encounter difficulties? How could you tell when you were making progress? How could you tell when you had solved the problem? What were your feelings during the problem solving process? How did it feel to have solved the problem?

Debrief in triads. Get together in groups of three. If you don't know the members of your triad, introduce yourselves. Then, to the extent that you care to, share your problem examples. (Some problems might be of a personal nature, and the participant may not want to share specific details.) Those of you who have been trained in active listening techniques may want to use these active listening techniques. Active Listening is a technique that is useful in helping to solve a variety of person-to-person communication problems. An Appendix on Active Listening is given near the end of this booklet.

Purpose of the exercise. [Note to reader: Explanations such as this are intended for the reader, and they are usually not included in the workshop presentation. A process-oriented exercise creates a certain ambience, and an "intellectual" discussion of such an exercise tends to destroy the ambience.] This opening exercise serves three purposes. First, it gets all workshop attendees immediately involved as participants. Second, it helps make the workshop participants feel good about themselves. If you recreate in your mind something that you have done well and feel good about, it will tend to make you feel good. If you give people a chance to share and talk about a past success, this will add to their good feelings. It also feels good to begin to fit into a group--to be a participating and sharing member of a group. Third, the exercise gets workshop participants thinking about problems and problem solving. Each now has in mind a specific problem and a process that solved the problem.

A Definition of a Formal Problem

Every person in this workshop encounters and copes with a large number of problems every day. Many of these problems are routine and solving them becomes almost automatic. But think for a moment about the variety of problems you work with in a typical day on the job…. This should convince you that you are an accomplished problem solver and know a great deal about problem solving.

Problem solving has been carefully studied by many great thinkers. There are a number of books that define the concept we call problem and explore a variety of problem solving techniques. A short bibliography is given in the References section of this booklet. We will use the following four components as a definition of problem.

1. Givens. There is a given initial situation. This is a description of what things are known or how things are at the beginning.
2. Goal. There is a desired final situation (or more than one). This is a description of how one wants things to be, a description of the desired outcome.
3. Guidelines. This is a listing or description of the general types of steps, operations or activities one can use in working to move from the Givens to the Goal. Guidelines are the resources, the facilities, the powers of the problem solver.
4. Ownership. In order for something to be a problem for you, you must accept some ownership. You must be interested in solving the problem or agree to work on the problem.

[Note to reader: The choice of vocabulary—Givens, Goal, Guidelines—is for the mnemonic value of the three G's. Other writers may use different vocabulary. When we say that a problem is well defined, we mean that the three G's are clearly and carefully specified. A well-defined problem can be worked on by people throughout the world over a period of time. Progress toward solving the problem can be shared, and cumulative progress is possible. In my opinion, this is one of the most important ideas in problem solving.]

We frequently encounter problem-like situations that have some, but not all, of these four components. We will call these problem situations. Often the most important step in solving a so-called problem is to recognize that it is actually a problem situation and then do the work necessary to obtain a carefully defined problem. This requires careful thinking, drawing on whatever knowledge one has that might pertain to the problem situation. Often a group of people will have a brainstorming session to get relevant ideas. (See the works of J. Pansey Torrance listed in the References. His research and development group has produced instructional material designed to help students gain improved problem solving skills. Also see books by Edward de Bono.)

Each of the four components may require further explanation in order to become clear to you. We begin with the last one: Ownership. Some experts on problem solving exclude this component, while others give it considerable weight. If coping with a particular situation is essential to your survival, you are apt to have considerable Ownership of this situation. But if the situation is a hypothetical (school book) exercise of little intrinsic interest, you may have little or no Ownership. Ownership is a mental state, so it can quickly change.

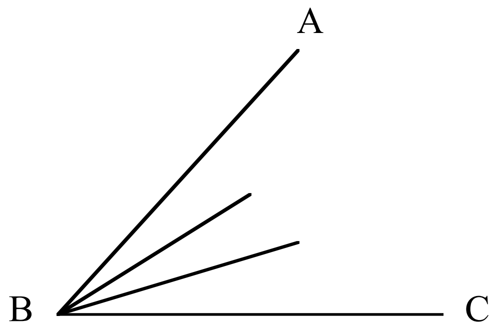
The issue of Ownership is particularly perplexing to educators. They recognize that Ownership—deep interest and involvement with a situation—often contributes to deep and lasting learning and intellectual growth. Thus, teachers often expend considerable effort to create situations that will get their students to have Ownership.

The alternative to ownership is coercion. Keep in mind that problem solving is a higher-order mental activity. Most people do not perform higher-order mental activities well under coercion.

As an aside, you may know some students who have spent literally dozens or even hundreds of hours working on a particular computer program or mastering a particular computer system. You may have said to yourself: "If only I could get all of my students that deeply involved." It is clear that such Ownership of a computer-related problem has changed the lives of a number of very bright and talented students.

Many workshop participants are, at first, puzzled by the Guidelines component of the definition of problem. Suppose you were giving your students a spelling test. From the student viewpoint, the task of correctly spelling a word is a problem to be solved. The student would be successful if allowed to use crib notes or a dictionary. What makes the problem a challenge is that these aids, and other aids such as a neighboring student's paper, are not allowed. The Guidelines specify that the students are to do their own work, not making use of crib notes or a dictionary. Note that Guidelines are often implicit, rather than explicitly stated. Confusion sometimes results because Guidelines are not explicitly stated.

For the mathematically oriented reader, another excellent example is provided by the problem situation of trisecting an angle. The problem situation is that one is given an arbitrary angle to be trisected. In the figure below, angle ABC is an arbitrary angle (that is, it is of unspecified size). The goal is to do a geometric construction that divides angle ABC into three equal angles.



Trisect Angle ABC.

Sometimes the Guidelines specify that one is only allowed to use a straight edge, compass and pencil. In that case it can be proven mathematically that the problem cannot be solved. In other cases one is allowed to use a protractor in addition to the other implements. Then the problem is easily solved by measuring the angle, dividing the number of degrees by three, and constructing new angles of the resulting number of degrees. Notice in this latter case the compass is not used, even though it is available. This is a good example since it demonstrates both that not every well-defined problem can be solved and that not all of the available resources must be used in solving a problem.

For a third example, consider the problem situation that teachers in a particular school seem to be making substantial use of pirated software. One can investigate the problem situation to clarify the Given situation—that pirated software is being used by teachers. One can set a Goal, such as reducing the use of pirated software by two-thirds in the first year, and decreasing it still more the second year. As a responsible and ethical educational leader, you may have considerable Ownership of the problem situation. But what are the Guidelines?

Optional Exercise. If time permits, this is a good place to do brainstorming and/or sharing on Guidelines for the software piracy problem situation. Workshop participants can share ideas on what they have done in their schools or what they think might work. This activity helps illustrate the difference between working to define a problem (get all four components clearly satisfied) and actually solving a problem. The brainstorming and sharing can produce a list of possible actions that one might take to solve the problem. But there is no guarantee that these actions can actually be taken in a particular school situation, or that taking the actions will solve the problem.

The piracy problem situation also illustrates a different but important aspect of Ownership. Many of the suggested Guidelines will involve changing the behavior of teachers in the school. If they have no Ownership, they won't be supportive of implementing steps that might resolve the problem situation.

Often a problem situation will lack a clear statement of the Givens and/or Goal. For example, we might have begun our discussion of software piracy with the statement: "I am a school principal. We have a software piracy problem in our school. Many teachers are using pirated software, and they let their students make copies of this pirated software."

According to our formal definition, this is a misuse of the word problem. While Ownership is implied in the statement, there is no stated Goal and no suggested Guidelines. Thus, only a problem situation has been described. One can work on solving or resolving a problem situation. Typically quite a bit of the initial effort will be expended in more carefully defining the situation—that is, more clearly identifying the Givens, Goal and Guidelines. One extracts from the problem situation one or more well-defined problems and then works to solve the well-defined problems.

Exercise. Please return to the problem you thought of at the beginning of this workshop. Check it against each of the four major components in our definition of problem. Does your example contain all four components? Be aware that some may be implicit in the situation you had in mind, and not explicitly stated. For example, Ownership is inherent to the fact that you solved the problem and remember a number of details about doing so. Guidelines include using the resources of your mind and body.

Whole Group Debrief. A Whole Group Debrief (we usually will just call it a Debrief) is an opportunity for sharing among all workshop participants. Someone may have had a neat idea that they want to share. Someone may have suggestions for classroom applications. People may have questions that they feel need to be answered or discussed. Remember, a workshop is a blend of content and process.

Applications

The overriding goal of this workshop is to improve the quality of education being received by students in our schools. This will occur to the extent that ideas presented in the workshop are truly important, and that workshop participants integrate them into their own educational work settings and behavior.

Some of the exercises, definitions, and ideas from the workshop can be directly used with a wide variety of students. Mainly, however, individual teachers will need to develop their own ideas as to what is important and relevant to their own students. A short, one-shot workshop (such as this one) can plant seeds for classroom and curriculum change. But whether the seeds grow and flourish is up to the specific participants in the workshop.

Research on effective inservice suggests that one-shot inservices are not very effective in producing change in the classroom. I would like this workshop to be 100 percent effective. I will define the workshop to be effective for a participant if the participant uses at least one idea from the workshop in his/her work within the next month. Please do not leave the workshop without one such idea in mind. Do your share in making the workshop 100 percent effective!

This is not intended as a curriculum development workshop. However, if time permits it is quite appropriate to discuss classroom applications. If the workshop consists of a homogeneous group (for example, a complete workshop of fifth grade teachers) then a whole group discussion on applications might be appropriate. If the participants come from a variety of grade levels and subject areas, it is appropriate to divide into small discussion groups.

A few suggestions for classroom activities are given below. If you are a teacher, you should have little trouble thinking of several applications relevant to your own teaching situation.

1. Have members of your class make a list of examples of problems. Write the list on the chalkboard; do not comment on the quality or characteristics of the problems your students suggest. After an extensive list is gathered, have students point out

common characteristics. Likely you will get three of the defining characteristics of a problem (Givens, Goal, Ownership). A little probing will likely lead to your students also providing examples of Guidelines.

Consider the example: "I only had a dollar, and I wanted to buy my mother a nice present." Ownership is evident in the use of the words "I" and "my mother." The Given situation is having a dollar, while the Goal is to have a nice present. The stated Guideline is that the present is to be purchased. Other possible methods for obtaining a present are excluded. Such exclusion may not have been intended. The student may go on to explain purchasing some materials that were then used to repair mother's favorite old purse, and presenting the purse as a present.

After all four general characteristics of problems are discovered, work with the class to see which of the problems proposed by the students have all four characteristics. Avoid making judgments. Rather, suggest that often some of these characteristics are not explicitly stated, so that one must seek them out. Learning to play the game of seeking out the hidden parts is a very important aspect of problem solving.

2. Make a large wall poster (you may want to use computer graphics software to print out the poster) that lists the four defining characteristics of a problem. As you talk to your class about whatever you happen to be teaching, pay attention to your use of the word problem. During the first few days when you use the word, point to the poster and explain how your use of the word fits all four characteristics. After a few days, merely point to the poster when you use the word problem. After a few more days, a modest head nod in the direction of the poster will suffice. The goal is to increase students' awareness that problem solving is a routine part of everyday life and of every academic discipline.

3. Select some categories of problems for a bulletin board display. You might select such categories as:

   a. Problems faced by various levels of government (school, city, county, state and federal).

   b. Problems shared by doctors, lawyers and others facing increased insurance costs.

   c. Problems faced by people traveling to foreign countries.

   d. Problems faced by members of minority groups.

   Have students bring and post newspaper and magazine clippings of headlines or short articles that discuss problems fitting into these different categories.

4. Discuss the concept of a well-defined problem with your students. The basic idea is that the three G's in a well-defined problem can be communicated to other people so that they can work on solving the problem. Have each student make up a well-defined problem and then communicate it to someone else in the class. Another approach is to have one person communicate his/her problem to the whole class and have the class discuss the problem.

   This can be a powerful and instructive exercise, and it can be used more than once. Consider the following "problems" suggested by two students:

a. I have a five dollar bill, three one-dollar bills, and five quarters. How much money do I have?

b. My problem is that my sister keeps borrowing my clothes.

The problem defined in A has one correct answer (although it can be represented in different ways). Thus, it can be solved by a class member, and the answer can be given to the person posing the problem.

The problem situation stated in B can be brainstormed by the class. But who is going to implement one or more of the suggested ideas? Do we have any guarantee that implementing the suggestions will actually resolve the problem situation?

5. Discuss with your class the idea that most so-called problems are actually problem situations. Often the missing ingredient is the Guidelines--a person just can't think of things they might do that might possibly lead from the Givens to the Goal. Brainstorming (individually or in a group) is a useful method of generating Guideline ideas. Select a problem situation suitable to the level and interests of your students. Lead your class in a brainstorming session. (Brainstorming is a major theme in books by Edward de Bono.)

6. Read through the activities given below. Select one that can be modified to fit your teaching situation. Try it with your students.

Activities

We conclude each major part of this chapter with a few activities that might be used to test and/or expand your knowledge of the materials just presented. If this booklet is being used in a course requiring homework assignments, the activities fulfill these requirements suitably.

1. Name three different academic disciplines in which you have some interest. For each, specify a problem. Notice that varying levels of Ownership are possible. Also, be aware that you are not being asked to solve the problems or even to assert that you know how to solve the problems. The intent is to increase your awareness that each academic discipline is concerned with carefully defining and working to solve particular categories of problems.

2. "In the United States during 1985, about 45,000 people were killed in motor vehicle accidents. That is a serious problem." Actually, this statement is a problem situation, with an inherent suggestion that the number of motor vehicle deaths that might possibly occur sometime in the future should be reduced by some (unspecified) actions to be taken by some (unspecified) agent. Make up three different well-defined problems from this problem situation.

3. Give two examples of problems than have not yet been solved, but which you feel may eventually be solved. Give two examples of problems that cannot be solved. Explain why each of your examples has the required characteristics.

4. Often people attempt to distinguish between real-world problems and academic or non-real-world problems. Explain ways one might tell a real-world problem from other problems. Do you feel this distinction is useful? Why, or why not?

5. Some writers, especially those writing about mathematical problem solving, distinguish between an exercise and a problem. An exercise is (a problem or problem-like thing that is) the same as or nearly the same as something one has encountered before. Thus, students will be shown a technique for long division and then be asked to do a number of long division exercises. Suggest some arguments for and against attempting to distinguish between an exercise and a problem.
6. We began the workshop with an indication that one of its purposes was to explore the problem of roles of computers in problem solving in education. Is this really a problem, or is it a problem situation? Justify your answer.
7. If a problem is sufficiently well defined (i.e., if the three G's are carefully specified), then a number of people throughout the world can work on the problem over a period of time. A medical problem such as AIDS provides an excellent example. Give three additional examples of problems or problem situations that are sufficiently well defined so that a number of people throughout the world are currently working on them and are sharing their results.
8. Name a problem which is very difficult for computers to solve, but which computers can solve. Explain why the problem is difficult for computers to solve. Then name a problem which you are quite sure a current computers cannot solve, and explain why they cannot solve the problem. Do you think that computers in the future (50 or 100 years from now) will not be able to solve the problem?

Part 2: Key Ideas in Problem Solving

What is Important in Problem Solving?

We each have considerable skill in problem solving. Some of our skill is based on what we learned in school, while quite a bit was gained in the "school of hard knocks." We each have our own thoughts on what are the most important ideas in problem solving.

Exercise. I would like you to think about some of the things you know about problem solving. You may want to write a brief list. [Note to reader: These instructions are deliberately vague, with no examples. The intent is to see what types of ideas the workshop participants come up with. I am generally surprised by the wide variety of responses.]

> As your list expands, begin to think about which of the ideas are most important from your point of view. That is, suppose that you were working with a group of students and you could have 100 percent success in teaching them two or three key ideas about problem solving. What would you have them learn?

Deb**rief.** The way I debrief this exercise is by building a list of ideas on an overhead projector. I begin by asking for a volunteer to share one idea. I write it down without comment (but ask questions for clarification if necessary). I ask for a show of hands for how many people listed that idea. Then I ask for another volunteer, and so on. I continue until the workshop participants run out of ideas or I get tired of writing.

When I do this exercise in a workshop, I indicate that I have made my own personal list of three important ideas. I suggest that I will be most impressed if the workshop participants are able to guess all three items on my list. What follows is a composite list from several workshops. It contains part of my response as to the three most important ideas. Read through the list. Add your own ideas. See if your ideas coincide with mine (which are given later).

1. Tenacity. If at first you don't succeed, try, try again. P.S.: If you still don't succeed, rethink the problem and the approach. Try a different approach.
2. Use your time effectively.
3. Transfer skills and knowledge—draw upon related knowledge and ideas from areas that may not be the same as that of the problem at hand.
4. Understand the problem.
5. Think about and try to find a variety of approaches or possible solutions.
6. It is okay to try and not succeed. Failure is one aspect of problem solving.
7. Keep the goal in mind—find the answer.
8. In problem solving (in a school environment) it is the process, not the answer, that is most important.
9. Break the problem into manageable pieces.
10. Maintain your self-confidence.
11. Seek out appropriate data that might be useful in solving the problem.

12. Keep an eye on the resources available to you.
13. If the problem involves a number of variables, consider one variable at a time.
14. Look for patterns. Organizing the data into a table may help.
15. Make a simpler example and try to solve it.
16. Draw a picture or a diagram.
17. Plan ahead; mentally try out your ideas before doing a lot of work actually using the ideas.
18. Use brainstorming techniques. Don't get stuck in a rut.
19. Respect your hunches.

It is fun to analyze the responses and to make guesses about the people providing the responses. For example, I would guess that the person providing response 13 is a science teacher. Science teachers tend to think about the variables in a laboratory experiment and how changing one variable may affect an overall situation.

Response 7 emphasizes finding the answer. An emphasis on finding the answer suggests to me a person who thinks of a problem as having only one right answer. This is a very narrow (and usually incorrect) viewpoint. Perhaps the person suggesting response 7 is thinking of arithmetic computational problems and is more interested in product than in process.

Response 8 also mentions finding the answer, but it focuses on the importance of process. A focus on process is particularly important when studying problem solving and practicing solving problems. One of the major goals of a school is to provide a safe environment in which students can experiment or practice with different processes for attempting to solve problems. Thus, perhaps this response was provided by an academically oriented person. Response 8 fits many academic problems, but doesn't fit very well with many real-world problems. In real life one often must come up with some course of action, since failure to decide is a form of deciding.

Response 9 is one of many well-known problem solving techniques. It is strongly emphasized as a useful approach in writing computer programs (top-down analysis). Thus, the person providing this response may well be a computer science teacher.

Responses 14, 15, and 16 may have come from math teachers. These are techniques taught by many math teachers. There are quite a few math curriculum materials that are designed to help teach such techniques.

Response 17 lies at the very heart of rational problem solving. In what follows we have chosen to assume that this idea is being followed in all problem solving processes and activities. Thus, we do not focus on it as a key idea needing further discussion.

Response 18 suggests brainstorming. Many books on problem solving emphasize and teach brainstorming techniques. The authors of such books often talk about "creative" problem solving. They suggest that creativity can be learned, and they provide lots of exercises designed to increase creativity. The works of de Bono and Torrance are especially noteworthy in this regard.

Moursund's Three Key Ideas

Each of the ideas suggested by the workshop participants is important, and it would be easy to extend the list. A comprehensive book on problem solving would cover all of the suggestions, and more. But the purpose of this booklet is to understand roles of computers in problem solving. This can be done by studying a small number of key ideas. My suggestions for the three most important ideas in problem solving are given below.

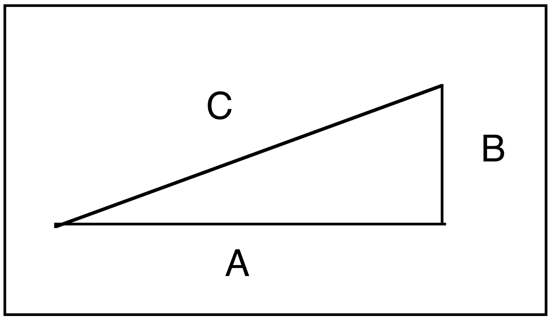
1. Understand the problem.

   This requires both general knowledge and skills (reading, writing, arithmetic, speaking, listening) and specialized knowledge in the disciplines of the problem. Generally speaking, people who are communicating problems assume such general and specific knowledge and skills. A broad-based, liberal arts education is quite useful background for solving problems.

2. Build on previous work of oneself and others.

   Suppose that it were not possible to build on the previous work of oneself and others. Then each problem encountered would be an entirely new experience, requiring starting from scratch. Each person would have to "reinvent the wheel" for every problem encountered. Cumulative progress would be impossible.

   The following example illustrates cumulative progress. What do the formula $C^2 = A^2 + B^2$ and the diagram of a right triangle bring to mind?

   You probably thought of a high school geometry class and/or the Pythagorean theorem. The language and notation of mathematics are very precise. They allow communication over time and distance. Pythagoras lived in Greece more than two thousand years ago. We have made many thousands of years of cumulative progress in mathematics. Much of this progress is inherent to the notation and vocabulary of math instruction, even at the elementary school level. The inventions of zero and the decimal point were major mathematical achievements.

3. One's problem solving skills can be improved.

   It is generally understood and accepted that the explicit study of problem solving and devoting considerable time to practicing solving problems leads to improved problem solving skills. The research literature in this area is extensive. A comprehensive literature survey is contained in Fredericksen (1984). The References also give a number of books designed to teach problem solving.

   As an example, consider the following steps that one might follow in resolving a problem situation:

   1. Study the problem situation to understand why it is not a well-defined problem. That is, determine which of the four defining characteristics (Givens, Goal, Guidelines, Ownership) are missing or not sufficiently clear.

2. Determine a well-defined problem that you feel appropriately represents the underlying intention in the problem situation. (You may find that several different well-defined problems arise from a single problem situation.) This may require considerable creativity, and brainstorming techniques may be useful.
3. If you are able to do so, solve the problem you have defined and proceed to step 5. If you are unable to solve the problem, proceed to step 4.
4. Keep trying to solve the problem you have defined. If you succeed, proceed to step 5. If you are unable to solve the problem, return to step 1 and/or 2. (Or, you might eventually decide that you are unable to resolve the problem situation by this approach. In that case, quit. Be aware that not every problem situation can be resolved and not every well-defined problem can be solved.)
5. Determine if the problem solution you have obtained is an appropriate and adequate resolution of the original problem situation. If it is, you are done. If it isn't, return to step 1 and/or step 2. (Or, you might eventually decide that you are unable to resolve the problem situation by this approach.)

These five steps can be memorized and repeatedly practiced with a wide variety of problem situations in a wide variety of disciplines. Eventually their use becomes second nature. Research suggests that a person who regularly follows this five-step process is apt to be a better problem solver than one who doesn't.

Of course, one can argue that people should not be trained to approach problem solving through this five-step approach. Perhaps there are better approaches for students to learn, or perhaps no explicit approaches should be taught to students. The latter may be a philosophical issue that can only partially be resolved by educational research.

In recent years there have been a number of articles which discuss lower-order versus higher-order skills. (Problem solving is a higher-order skill.) These articles tend to stress that over the past decade or so, schools in the United States have placed increased emphasis on lower-order skills. As a consequence, national assessment scores on lower-order skills actually increased. However, this came at the expense of a substantial decline in test scores on higher-order skills. Now educational leaders are calling for renewed emphasis on higher-order skills (A Nation at Risk, 1983; Beyer, March 1984; Beyer, April 1984; ERIC, 1984).

Exercise. Please raise a hand if you have had a formal course on problem solving. How many of you have attended a workshop on problem solving (not counting this workshop)? Would one or two of you please share the nature of the problem solving course or problem solving workshop you attended? In what ways was the experience beneficial to you?

Debrief. A few colleges and universities give general-purpose, interdisciplinary courses on problem solving. Workshops on problem solving are fairly common, although they often focus on problem solving within a specific discipline. This is unfortunate, since most real-world problems are interdisciplinary in nature, and many problem solving techniques are generic. Usually only about 10% of the participants in my problem solving workshops have had a formal course or an extended workshop on problem solving. Once in a great while I encounter a teacher who is teaching a course in problem solving in his or her school.

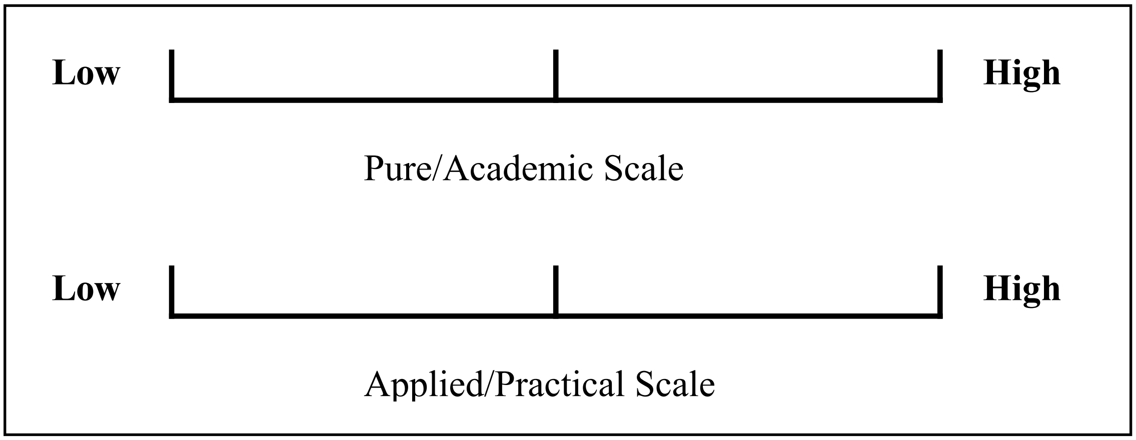
Exercise. Please bring to mind a problem that you have recently encountered and solved. It can be the one you thought of at the beginning of the workshop, or it can be a new problem. Do a quick mental check to make sure that it satisfies our formal definition of a problem. Then carefully think through the process you used in solving the problem. Does the process you followed support the first two of Moursund's key ideas on problem solving?

Debrief in triads. Share your example and thinking with others in your triad. Practice arguing for or against Moursund's first two key ideas. For example, you might argue that some other idea is more important than these two ideas.

Debrief in whole group. Does someone have an important idea they would like to share with the whole group? [Note to reader: Here I sometimes have difficulty in avoiding an argument. If someone suggests an idea they feel is more important than mine, I acknowledge their idea and indicate that is why so many books have been written about problem solving. I sometimes even argue with myself. The idea of mentally contemplating possible outcomes from different approaches to solving a problem is certainly of utmost importance.]

Purpose of the previous two exercises. We will discuss Moursund's three ideas more in the next part of the workshop. Here the intent is to understand the ideas—to begin thinking about whether they really are important, or whether there are clearly some other, more important, ideas.

Exercise. I want you to consider two scales (see diagram below). One scale is labeled Pure/Academic, while the other is labeled Applied/Practical. We all know people who are a whiz at schoolwork—who can solve academic or textbook problems with ease. Sometimes such people seem to have little talent in coping with the problems of life outside the sheltered environment of the educational world. (Some university professors are accused of falling into this category.)



Conversely, we all know people who seem to cope beautifully with real-world problems, even though they have little formal education or did quite poorly while in school. Such people are sometimes said to be "street wise" or "street smart."

Now I want you to label the four end points of the diagram scales, using people that you know. For example, the left end of the Pure/Academic scale would be labeled with the initials or name of a person you know is very poor at coping with school-type problems.

Finally, I want you to place yourself on each of the scales.

Debrief in Triads. To the extent that you are willing, share your feelings and thoughts on this exercise with the others in your triad group. Did you learn anything about yourself by doing the exercise? Are you happy with your relative positions on the two scales? Have your positions changed over the past five years? Are your positions apt to change over the next five years?

Debrief. I'd like a show of hands. How many people ranked themselves above average on both scales? (In the workshops where I have used this exercise, most participants rank themselves above average on both scales. That isn't too surprising, since all are college graduates.) Next, I'd like a show of hands on how many people ranked themselves higher on the Pure/Academic scale than on the Applied/Practical scale. (Typically, the majority of participants rank themselves higher on the Pure/Academic scale.) Finally, is there anybody who learned something from this exercise that they would like to share with the whole group? (I often get a response here indicating increased awareness that much of what goes on in school may be quite far removed from the real world of many students. This is a response I am looking for, as it is a key point in a later part of the workshop.)

Purpose of the exercise. Up to this point, I have deliberately hidden what I consider to be the main purpose of the exercise. Our schools make a significant effort to help students gain problem-solving skills that will be applicable in both academic and real-world settings. But good transfer of learning often does not occur. Frequently the real world seems quite far removed from the world of school. Indeed, we all recognize that even transfer between somewhat similar academic disciplines (for example, math and the physical sciences, or sociology and political science) seems to cause students a great deal of difficulty.

The purpose of the exercise is to get workshop participants to begin thinking about the informal, non-school, learn-by-doing type of learning that occurs so naturally for almost all people. One of the key ideas in Seymour Papert's *Mindstorms* book is that such learning, when it occurs, is both rapid and fun, and should be encouraged. It is human nature to learn, and we are all natural-born lifelong learners. This ties in well with the next part of the workshop, which is an analysis of possible effects of computers on the three ideas that I think are most important in problem solving.

Applications

1. Have your students select some problems they have previously solved, and have them mentally review the steps they followed in solving the problems. Then have students give examples of the process steps they followed in solving their problems. Write examples on the chalkboard without commenting on them.

   When you have collected a large list, work with the students to categorize the results. Part of the exercise is to develop suitable categories. For example, a student might have suggested: "At first I couldn't do it, but I just kept trying and trying." This provides an example of tenacity. Another student might have suggested: "I got stuck, and I asked my mother for help." This provides an example of building on the work of others (previous knowledge gained by mother).

2. As a continuation of the above exercise or as a new one, have each student get firmly in mind a problem s/he has recently solved. Put on the chalkboard the two key ideas:

   a. Understand the problem.

   b. Build on previous work of yourself and others.

   Ask several students to share how their problem examples illustrate these two key ideas. Have students mentally check their problem solving processes against these two ideas. Make a poster containing these ideas. Post it in your room and refer to it frequently until the two ideas become second nature to your students.

3. As a continuation of the above exercises or as a new exercise, work with your students to identify the discipline or combination of disciplines that provide information or skills needed to understand some specific problem. (Have several different problems available from several different disciplines for use in this activity.) The idea is to emphasize the interdisciplinary nature of problem solving and that each discipline has problems that it considers.
4. Have your students make personal lists of ideas or information that they learned a long time ago and which they now frequently use when solving problems. The idea is to get your students to think about how they build on previous work of themselves and others when solving problems. This may lead to a discussion of such basics as reading, writing, arithmetic, speaking and listening.
5. How can one tell if two problems are the same or nearly the same? For example, tying my left shoe is nearly the same task as tying my right shoe. I don't have to learn different procedures for tying left and right shoes. Discuss this question with your students. Then have each student make a list of problems that are nearly alike, so that the same general type of solution procedure can be used to solve all of the problems in a student's list.

   This exercise illustrates a very important idea. We help students learn to solve important categories of problems. But then we also need to help students learn to recognize problems that belong to the categories they know how to solve. This problem-recognition task can be quite difficult. For example, a student may learn to handle metric measurements in a math class, but be totally unable to deal with metric measurements in a science class. The idea of transfer of learning should be carefully considered in all problem solving instruction.
6. Start an "A Problem a Day" assignment in a course you teach. Every day each student is to write a brief description of a problem encountered outside of the particular class (topic, subject) you are teaching. The problem must have the characteristic that working on it makes some use of the material from the class you are teaching. The problem can come from another class or subject, or from outside of school.

   This is another exercise focusing on transfer of learning. We want students to think about the applicability of what they are learning. For example, perhaps students are in a literature class and are studying Shakespeare. They are, of course, improving their general cultural background. But Shakespeare was a keen observer of human nature. Many of the brief quotations that people remember from his writings are comments about important problem situations that people encounter. Or, one can look at Shakespeare from the viewpoint of his influence on the English language. A number of our everyday words/phrases can be traced back to his writings.

Activities

1. Select six of the Important Ideas from the list suggested by workshop participants earlier in this part of the booklet. For each, explain what you think the workshop participant had in mind. Then make some guesses about the nature of the workshop participant who suggested each problem solving idea. Many of the suggestions are relatively interdisciplinary, but quite a few are likely to originate in specific courses.

The idea is to increase your conscious awareness of where students might encounter key ideas on problem solving.

2. Argue both for and against the suggestion that the main difference between a school problem and a real-world problem is Ownership.

3. School can be viewed as a safe place to practice solving problems. There one can try, and not succeed. In the real world, however, not succeeding can have serious consequences. Discuss the extent to which you and/or our school system provide a safe and supportive environment in which students can practice solving problems. Your discussion might include your thoughts as to whether school should provide such a safe environment. Many schools are quite competitive and harsh, and they may not provide such a safe environment.

   The issue of cooperative learning might also enter your discussion. Research on cooperative learning is strongly supportive of this practice. Similarly, research on cooperative problem solving in a school environment is strongly supportive of this practice. But, does such cooperative learning and cooperative problem solving adequately prepare a student for life outside of school?

4. Most lists of the goals of an educational system include a statement about problem solving. In recent years, many national reports have suggested that our schools should be doing better in teaching problem solving. A counterargument is that the home and other non school environments are critical sources of instruction and practice in problem solving, and that declining problem solving scores reflect changes in home environments. Develop a strong case for each side of this debate, and then indicate your personal opinions on the issue.

5. The Pythagorean theorem example given earlier illustrates precise communication over time and distance. Give some examples from areas outside of mathematics in which such precise communication occurs. Keep in mind that often the purpose of the precise communication is related to problem solving. If your examples relate to solving particular categories of problems, identify the categories of problems.

   For one possible source of examples, consider oral tradition and parables. A parable generally contains an important message about how to handle some type of problem situation.

6. Toffler's book *The Third Wave* traces an orderly historical flow from hunter-gatherer era to agricultural era to industrial era to our present information era. Think of the ideas of building on previous work of others, and of communication over both time and distance. During the past 150 years there has been a marked improvement in speed, ease and reliability of communication (telegraph, telephone, computer-based communication systems, etc.). Discuss the emergence of the Information Age as a natural byproduct of improvements in communication. Project the continuing improvements in communication somewhat into the future. How should such improvements in communication affect education?

7. Many real-world problems are interdisciplinary in nature, and solving them requires using skills and knowledge from a variety of disciplines. However, our school system is very discipline-oriented. For most students, the school day is broken into

distinct pieces in which specific disciplines are studied. Discuss why this is so and how it affects the teaching and learning of problem solving.

Part 3: Roles of Computers in Problem Solving

Understand the Problem

In this part of the workshop we will examine the three key ideas from the previous part, exploring them in order to better understand various roles that computers play. We begin by addressing some issues involved in understanding a problem.

We have defined a problem to have four components: Givens, Goal, Guidelines, and Ownership. In the remainder of this part of the workshop, Ownership will be assumed. To "understand" a problem means to have a functional understanding of the three G's defining the problem. Thus, information about the three G's serving to define the problem must be in one's mind/body, and one must have a reasonable ability to work with this information.

Often much of the information helping to define a problem will be available in written or oral form. This is particularly true in academic problem solving situations. Thus, one needs to have reading and listening skills to be able to access information helping to define a problem. Frequently one makes use of speaking or writing to seek out additional information about a problem. My conclusion is that the basics of education (such as reading, writing, arithmetic, speaking and listening) are very important in understanding a problem.

More generally, one may make use of any and/or all of one's senses to obtain information helping to define a problem. Thus, learning to use one's senses is an important aspect of learning to solve problems.

I don't intend to go into detail on what it means to understand the information that helps to define a problem. However, it seems evident that we often represent such information using words, sounds, and other symbols. For example, a chemistry student might encounter the following symbols:

$$H_2SO_4 + 2NaOH \text{ --> } Na_2SO_4 + 2H_2O$$

This is a sentence written using symbols that allow chemists to communicate precisely with other chemists over time and distance, and across cultures. Throughout the world, students in chemistry classes learn how this combination of an acid and a base yields a salt and water.

Triad Group Exercise. Think of an area in which you have considerable specialized knowledge. Pick an area in which you believe the other members of your triad probably don't know as much as you. Then give a brief explanation of your area to the members of your triad. Try to make considerable use of the technical terms, big words, and special notation of the area. What you hope is that you won't be understood, even though you would be communicating clearly to a specialist in your area.

Debrief. I observed a lot of laughing—I guess many of you enjoyed this exercise. Does anyone have an example to share with the whole group? What did you learn by doing this exercise? What implications does this have for education? (Note to readers: A variety of examples get shared. Many workshop participants have hobbies such as glass blowing, raising exotic pets, knitting and weaving, astronomy and so on. There are many ways people develop and exhibit their uniqueness. Education needs to provide for individual differences and encourage development of individual interests.)

Understanding the words, sounds, and other symbols educated people use to communicate may take considerable education and experience. Much of our K-12 educational system is designed to help all students gain a common core of understanding and experience. One can base arguments for liberal arts education on this type of analysis. Students throughout the country need to gain a common core of knowledge, skills, and experiences so they can effectively communicate with each other.

At this time, I want to point out two major ways computers enter this discussion. First, computers and related technology are a new body of knowledge. Many problems involve this technology. To understand such problems, one must know something about the vocabulary and basic ideas of computers. In this sense, computers make problem solving more difficult, because they expand the range of possible problems and the range of knowledge needed to understand problems.

This analysis supports a position that all students need to acquire a functional talking and reading level of computer literacy. This aspect of computer literacy should be oriented to understanding problems that involve computers and related technology. For example, computers can be used to create, maintain, and access large databanks of information about people. A number of serious social problems center around use and misuse of such databanks.

A second way that computers enter the area of problem understanding is through computer assisted learning (CAL). There is substantial research evidence and experience to support assertions that CAL can help many students to learn more, better, and faster. The most convincing evidence comes from studies of drill and practice software, especially when used to help students improve their lower-order skills. There is also good evidence to support use of CAL tutorials and simulations that are designed to help students improve both lower-order and higher-order skills. Thus, the use of CAL can have a major impact on problem solving, since it can help people more efficiently gain the knowledge needed to understand problems (Kulik, Bangert and Williams, 1983).

However, this is not a workshop on CAL. Thus, we will merely state our contention that the basics of education are essential in problem solving, and that computers do not decrease the need for a broad-based, liberal education. Indeed, I strongly believe computers increase the need for and value of such a liberal education. This education should be strongly interdisciplinary in nature.

Build on Previous Work of Oneself and Others

In my opinion, this is the area of problem solving that is most strongly impacted by computers. We will treat it briefly here, and then return to it in the next part of the workshop.

Exercise. I'd like each of you to spend a couple of minutes making a personal list of ways in which you build on previous work of yourself and others when you solve problems. Think of this as a brainstorming exercise, and write down whatever comes to mind. You may find that it helps to mentally review several different problems you have recently solved. Be aware that I am asking you to solve the problem of making a list of ways you build on previous work of yourself and others. You are solving this problem by drawing on ideas in your head—that is, by building on previous work you have done.

Pool your ideas with those of two other people. As you do this, you might note whether the way you have been thinking about this task is the same as the way others in your triad have thought about it. It is always interesting to learn whether one interprets directions in the same way as others. Be aware that the pooling of ideas is drawing on previous work of others.

Debrief. Let's build a list of some of the key ideas that have been developed in the triads. We will do this by accepting an idea from one triad, then another triad, and so on. After each idea is written down, we will have a show of hands to see how many triads encountered the idea.

A few ideas from workshop participants are listed below. This is not intended to be a comprehensive list. Rather, it is intended to give the flavor of typical responses that are discussed in triads.

1. I search my mind for whether I have run into the problem in the past. If so, I see if what I did in the past worked. If it did, I usually do it again.
2. When I am driving a car and a tight situation occurs, I react automatically. My body/mind knows what to do and it does it. The same things can be said about riding a bicycle.
3. I have a large file of exams I have given in the past. When I need to make up an exam, I look through this file and pull out questions for the exam.
4. I make a lot of use of the library. I am always checking out what others said about a particular topic.
5. I like to play the piano. I play music written by the 19th century classical composers.
6. I have a friend who knows a lot. I often ask this friend for help on my personal problems.
7. There are some things that I am sure I know how to do, such as arithmetic computations. As I think about solving a problem, I divide the solution process into chunks or pieces that I know I can handle. That way I can keep thinking about general ideas, rather than getting bogged down in the actual details of carrying out a solution process.
8. I play chess, and I spend a lot of time studying books of chess openings. This has improved my playing level.

A general theme is the storage and retrieval of information. The information may be stored in someone's head or it may be stored in print form. The total amount of stored knowledge is overwhelmingly large and is growing very rapidly.

Response 7 includes the idea of chunks—sub problems that one knows one can solve. This is a very important idea. The human brain is severely limited in the number of details it can keep in active consciousness at one time. It is easily overwhelmed by a novel or complex situation involving a large number of details. (Think back to when you were first learning to drive a car!) But the human mind can chunk information, and it can store kinesthetic processes in an automatic pilot part of the brain. The idea of chunks in problem solving is illustrated by the following. "In working on this problem I can see that I will need to do a lot of arithmetic and solve some equations. I know I can do those things, but they will take quite a bit of time and effort. Let me think about what else I will need to do to solve the problem."

Part 5 of this booklet addresses the idea of Effective Procedures, and it can be thought of as a more detailed discussion of certain aspects of chunking. We are particularly interested in effective procedures that can be carried out by a computer. Now we see a very important role of computers in problem solving. One can carry in one's head the information that a computer program exists that will solve a certain category of problem. This is a chunk—even if one doesn't know the details for writing the computer program. One makes use of these computerized chunks when actually carrying out a procedure to solve a problem.

More generally, we recognize that computers are a new aid in the storage and retrieval of information. Thus, they are a new aid in building on previous work of oneself and others. This is such an important idea that we will devote a major section of the workshop to it. (See Part 4 of this booklet.) But right now we will move on to the third key idea we want to discuss about problem solving and roles of computers.

Improving Problem Solving Skills

As noted earlier, problem-solving skills can be improved by a combination of explicitly studying problem solving and practice in solving problems. Your participation in this workshop, for example, constitutes explicitly studying problem solving and practicing solving some problems. Hopefully, it will make a contribution to your problem solving skills. You should be aware, however, that you have many years of experience and ingrained habits of problem solving behavior. A few hours of participation in a workshop represents only a modest contribution to your total training and experience. Thus, don't expect miracles!

Exercise. There is a rapidly growing collection of commercially available software designed to teach problem solving. Think of a piece of such software that you have used. Mentally review its use, and make a list of how using the software contributes to improved skill in problem solving.

Debrief in triads. Each person in the triad is to describe a piece of software and the process of using it. The description should focus on problem solving and how use of the software might contribute to improved problem solving skills. [Note to readers: Two examples are discussed below.]

Taxman, from MECC, provides a good example. Many workshop participants are familiar with this piece of software, since it has been in wide circulation for many years. In Taxman you begin with a list of integers, from 1 to a number you specify. You may select one of these integers from the list (except the number 1), and add it to your score. The computer (the taxman) then gets all integers from the list that are divisors of your integer, and these are added to the computer's score. These integers are removed from the list, and then it is your turn again. The game ends when you can no longer find an integer that has at least one divisor (other than itself) in the list. The computer then adds all remaining list elements to its score.

To play Taxman well, one must develop a successful strategy (which requires careful thinking or quite a bit of trial and error), and one must plan ahead. These are useful techniques in most problem solving situations. Playing the game well also requires that one do quite a bit of arithmetic, determining the divisors of various integers. Most people playing Taxman do the arithmetic mentally, thereby maintaining or improving their mental arithmetic skills.

The *Factory,* from Sunburst Communications, provides another excellent example of problem solving software. In this game one specifies a sequence of machines that can drill square or round holes in a part that is being manufactured. One also specifies machines that rotate the part and carry out other actions. The goal is to design a sequence of machines that will produce a specified part. One can also work to design a shortest possible sequence of machines to produce a specified part.

Students playing this game learn to plan ahead. They learn that a problem can be solved in more than one way, and that one solution may require fewer steps than another. They may gain in tenacity.

The Factory has another important value. In 1985 Pat McClurg, as a computer-in-education doctoral student at the University of Oregon, used this software in her dissertation research. The study was to examine improvement in geometrical visualization skills that comes from use of this and another game. Quite positive results were noted with a wide variety of girls and boys.

We have good evidence that geometrical visualization skills are important in learning to solve many kinds of mathematics problems. Thus, we might decide to have mathematics students play geometrical- visualization games in the hope that this will improve their mathematical problem solving skills. *Super Factory,* a three-dimensional version of The Factory, provides another example of software that might be used for such purposes.

You are all aware of computer simulations. In essence, a computer simulation defines a problem and provides feedback as a person attempts to solve the problem. For example, consider computerized flight simulators. These are so good that they can be substituted for a substantial part of the hands-on experience needed to learn to fly an airplane. More sophisticated flight simulators are used to train astronauts.

In essence, an educational simulation is good to the extent that moving from it to a real-world application is a near transfer. This is one way of evaluating such software. Think about what you want students to learn from using the software. Think about whether it is a near transfer to move from use of the software to working with real-world situations involving what you want students to learn.

The use of computer simulations in learning to solve problems is not widely implemented in precollege education. However, it is widely used in military and industrial training, in medical schools, and in other places where the cost of education is relatively high. It seems evident that it will be of increasing importance at all levels of education.

Exercise. Think of a computer simulation you have used with your students or for yourself. What problem was being addressed? (Was it a real-world problem or an imaginary problem, such as in a world of Dungeons and Dragons?) What did you learn by using the simulation? What skills did you gain? While you were using the software, did you think of ways the simulation could be improved? Did you think about near and far transfer?

Debrief. Is there anybody who would like to share an example with the whole group? I am particularly interested in examples where you are sure that you or your students gained increased problem solving skills. [Note to readers: An example is given below.]

The Oregon Trail simulation from MECC provides a good example for discussion because many workshop participants have used it. In Oregon Trail, one tries to travel from Missouri to Oregon City using an initial set of resources and facing a variety of difficulties. The simulation can be used with students over a wide range of grade levels. It has nice color graphics and maps.

It also has exciting action (hunting for food) and random events (both bad and good). It is reasonably accurate historically, and students tend to like playing this simulation/game.

But what do students learn by using the simulation? My observations are that student learning is quite limited unless a teacher provides substantial guidance. When I played the game, I learned that it was much easier to be rich than to be poor when traveling along the Oregon Trail. I learned that many people died during the trip. I also learned that I have poor and slow hand-eye coordination for handling the hunting part of this simulation/game.

The literature on effectiveness of computer simulations in precollege education (as contrasted with training) is rather sparse. This contrasts with quite a bit of solid literature on use of simulations in training situations, especially in industrial and military settings.

You are undoubtedly aware that there are many educational simulations that do not require use of computers. There is extensive literature on their use in schools. The results seem mixed and do not provide overwhelming evidence to support use of simulations. Some writers suggest the difficulty is that teachers who are used to a carefully controlled classroom environment and who use conventional fact-oriented tests are uncomfortable with use of simulations. They suggest this is particularly true in social science classes. (Tom Snyder has developed a lot of software of the sort we are discussing here. He gives excellent workshops that help teachers make effective use of these simulations in a classroom environment.)

In essence, we are into the issue of learning facts versus learning to think and to solve problems using the facts. It is relatively easy to test whether students have learned a collection of facts. And who can deny that factual knowledge is important in problem solving? The issue becomes, what is an appropriate balance between learning facts and practicing thinking (problem solving) using the facts?

Applications

1. Have your students use some drill and practice CAL materials. Then lead a class discussion on their reactions to such CAL. Did they feel they learned faster or better? Did they enjoy the experience? What are their arguments for and against increased use of CAL? Can they name situations in which they feel such CAL would be particularly useful? Can they name situations in which use of CAL would be counterproductive?

   This application relates to the idea of learning to learn. Students can think about what aids to learning are best for them. In my opinion, helping students learn how to learn is one of the most important goals in education.

2. Select a discipline that all of your students know something about. Lead a class discussion on what knowledge one needs to understand (but not necessarily to solve) some of the basic problems of that discipline. You might begin by asking students to state some of the basic problems addressed by the discipline.

   This application gets students to think of a discipline in terms of the specific problems addressed through the discipline. It emphasizes distinguishing between understanding a problem and knowing how to solve the problem.

3. Which is more important—learning facts or practicing thinking and problem solving using facts one has learned or can retrieve? Perhaps some disciplines are more fact

oriented while others are more problem solving oriented. Perhaps a fact-oriented education is better than a thinking-oriented education, depending on the social/political structure of the country in which one lives. Encourage your students to explore and discuss these ideas. Get them to tell their opinions and why they hold these opinions.

This application focuses on a very important issue. Our current educational system tends to reward students who have good and quick memories. Such students can quickly solve a wide range of relatively superficial problems through simple information retrieval techniques and glibness. Such success can get in the way of putting in the time and effort required to learn to attack more difficult problems. It has been my observation that many quite bright students fail to develop their intellectual potentials because of this.

4. Have each of your students select a problem that he or she has recently solved. Pair up your students. Each student is to carefully explain to a partner his or her problem and the steps followed in solving it. This gives students experience in clearly formulating problem-solving steps, and they may learn some problem solving ideas from each other. This can be a good exercise for giving your students instruction and practice in inquiry-oriented active listening. The discussion is to focus on the problem suggested by the initial speaker, and the listener isn't supposed to help solve the problem. But the listener might suggest related ideas and/or other approaches that might be applicable.

5. How do you know what you don't know? Make up a collection of problems, and read the problems to your students. For each problem, students are to indicate whether they think they can solve it. (Don't give them time to actually solve the problems.) Use the results to initiate a class discussion. The goal is to get students to think about knowing, not knowing, and how they (personally) can tell whether they know something or will be able to solve a certain type of problem.

6. Select a simulation/game suitable to the level of your students, and have them use it. (It need not be a computerized simulation/game.) Then have your students discuss and/or write about how playing the simulation/game relates to problem solving.

   This application is insidious. Perhaps it will lead to some of your students consciously thinking about problem solving as they play computerized arcade games or other games for recreational purposes.

Activities

1. Research evidence strongly supports the conclusion that most students can learn basic skills faster when conventional instruction is supplemented by computer-assisted learning. In recent magazine ads the Computer Curriculum Corporation (Patrick Suppes started the company in California in the late 1960s) has been claiming 100 percent learning rate gains in schools using its materials.

   Think about the problem situation of the claims and evidence of CAL versus the actual levels of use in our schools. Refine this into one or more carefully defined problems. Discuss some approaches that might help solve the problems you define.

2. Sometimes people discuss our educational system as being a simulation for participation in the real world. Viewed in this manner, what types of things might one do to make the simulation more like the real world, with an expectation of increasing transfer of learning and skills from the simulation to the real world?

3. Many books on problem solving emphasize the solving of puzzle problems. A typical example that comes to my mind is the farmer with a boat, a fox, a pile of grain, and a goose. Subject to various conditions, the farmer is to transport the fox, grain, and goose across the river, without getting the goose eaten by the fox or the grain eaten by the goose. Another example, known to many students studying use of recursion in programming, is the Towers of Hanoi.

   Think of an example of a puzzle-type problem you have studied in the past. Give a careful statement of the problem. (If you like and are able, you can also tell how to solve the problem.) Then explain how the study of this problem might improve a person's ability to solve real-world problems. Pay particular attention to the issue of transfer.

4. Select and learn to use a piece of problem solving oriented software. As you use the software, introspect about what skills, knowledge, and problem solving ideas you are using. Write a report on the merits of this software as an aid to improving one's problem solving skills.

Part 4: Accumulated Knowledge of Humans

A Model for Accumulated Knowledge

This part of the workshop focuses on roles of computers in building on previous work of oneself and others. To build on previous work, one must retrieve information about that work. Thus, a good starting point is to examine ways/places in which information is stored, and how these are affected by computers.

I have built a simple model that I use when talking about the accumulated knowledge of the human race. I divide the accumulated knowledge into three major categories. A few hundred years ago the three categories of this model were quite distinct. Progress in photography, audio and video recording, and computers, has blurred some of the distinctions.

1. Personal knowledge. This is the knowledge, skills and experience that individuals carry in their heads and bodies. Each person has unique knowledge, skills, and experiences. We can access some of another person's personal knowledge through use of our verbal and nonverbal communication skills. We can also hire people to apply their personal knowledge to a problem of interest to us.
2. Public knowledge. (If you like, think of this as published or sharable knowledge. At one time, one might have described this category as written and printed materials, drawings, and paintings.) I use this term in a very broad sense to include written materials, audio and video recordings, films and photographs, drawings and paintings, etc. It is information that can be transported over distance and preserved over time. It can be accessed by many people. Public knowledge is growing rapidly—I have heard and read estimates that it is doubling every 10 years or even more often.
3. Artifactual knowledge. There is a substantial amount of knowledge contained in artifacts we routinely use. Consider the overhead projector I use in workshops. It contains a light bulb, and it took the inventive genius of Thomas Edison to produce a practical electric light bulb. Or consider the optics of the overhead projector. The lenses and mirrors represent the thinking and skills of a number of early inventors and skilled crafts people. And, of course, the electricity that we tend to take for granted comes to us through the work of many scientists and engineers and a large supportive infrastructure.

   I realize that I am not using the word artifact in quite the way most people do. I want to include all objects, infrastructure and so on that we use. A crossbow represents artifactual knowledge, as does a steam engine or a printing press with movable type. Nylon stockings are an artifact—-hey represent and contain information on the chemistry of nylon as well as information on the art/craft of knitting A highway represents a considerable amount of information about engineering and materials science.

Exercise. Imagine life in a hunter-gatherer society, long before the invention of reading and writing. To what extent did the three types of knowledge discussed above exist at that time?

What constituted a good education for life in that type of society, and how did people obtain such an education?

Debrief in triads. The intent is to increase understanding that the nature of a high quality education changes with the nature of aids to storing, retrieving, and making use of accumulated knowledge. Even without reading and writing, there can be an increasing amount of knowledge accumulated in a society. For example, artifactual knowledge increases as better tools are developed and passed down from generation to generation.

In a hunter-gatherer society, most learning occurs through observation and imitation. Learning focuses mainly on skills and knowledge having immediate and continuing value to the learner. Children learn to make and use the tools (remember, tools represent artifactual knowledge) they will need to use in order to be productive adult members of their society.

Exercise. Imagine life perhaps four thousand years ago, in the agricultural era. Reading and writing are available, but only a select few have the opportunity to learn these skills. What changes in the amount and nature of the three types of knowledge might one observe in moving from a hunter-gatherer era into an agricultural era? How might this affect the nature of a high-quality education?

Debrief in triads. Pay particular attention to how reading, writing, and books contributed to and changed problem solving.

One way to solve a problem is to find out how someone else solved the problem, and then to do the same thing. Books make it possible to draw upon an ever-increasing collection of information on problems that have been solved.

Reading and writing are very useful in organizing one's ideas. They provide a supplement to the human brain as it works to solve a complex problem.

Each of the three general categories of accumulated knowledge enters into problem solving. To solve any problem, I draw on my personal knowledge. Without such knowledge I cannot understand a problem or take action to solve it. There are many problems that only I can understand and attempt to solve, since they involve personal knowledge that is completely unique to me. (One task of a psychotherapist is to help clients learn to solve their personal problems.)

Public knowledge is one representation of knowledge that is accumulated by artists, researchers, and scholars building on the work of previous artists, researchers and scholars A large library contains more information than a single individual can ever master. We all understand how the printing press and movable type (that is, inexpensive and widely distributed books) changed the world.

Perhaps the role of artifactual knowledge in problem solving is more subtle. I can solve many problems by making use of an artifact that was designed to help solve the problem. I solve the problem of feeling cold and wanting to not feel cold by putting on a sweater or by turning up the thermostat. I solve the problem of needing to travel between two cities by driving my car or flying in an airplane. I solve the problem of feeling hungry by eating a can of hash that has been stored on my kitchen shelf.

Notice that in all of these examples the nature of the previous education and experiences needed are different than a person would have needed a few hundred years ago. It is evident that artifactual knowledge, as an aid to problem solving, has a direct impact on the nature or type of education needed to cope with the problems encountered while living in a particular society. Fortunately, it often takes very little formal education in order to learn to make use of a particular type of artifactual knowledge. For example, one learns to drive a car mainly by doing it rather than by attending classes on driving or studying the underlying theory of automobile design and construction.

Computers and Personal Knowledge

As pointed out earlier, computers can be part of one's personal knowledge, and computers can be used to help gain personal knowledge. So far, the total impact of these two aspects of computers on people has been modest. Relatively few people include the subject of computers as a significant part of their total personal knowledge. Relatively few people have gained a significant part of their personal knowledge through use of CAL. But many students are enrolled in computer and information science courses, and the use of CAL is growing.

CAL can be thought of as a combination of public and artifactual knowledge specifically designed to help people gain personal knowledge. It seems evident to me that use of CAL will grow rapidly in many school systems. Through appropriate use of CAL it is possible for students to acquire factual knowledge and some basic skills more rapidly. This frees up time that can be spent on improving higher-order skills.

Moreover, there is a gradually growing collection of CAL materials focusing on higher-order skills. Earlier in the workshop we discussed educationally oriented computer simulations. These are a form of CAL that helps in the teaching and learning of higher order skills.

There is an increasing number of full-year, CAL-based courses available. Often these are designed so they can be used without the help of a teacher. As the quality of such courses improves and their cost declines, we can expect their use to expand.

Another interesting trend is combining CAL with artifactual knowledge for use in non school settings. Some machines (such as high end photocopiers) now come with built-in CAL systems that teach you how to use the machine. If a problem occurs when using the machine, you can switch into a CAL mode and the machine will help you learn how to cope with the problem. Perhaps you have heard of the concept of the teachable moment. This provides an excellent example of making use of a teachable moment.

One characteristic of our Information Era is steadily improving telecommunication systems. Such systems are making it easier and less expensive to communicate with other people. As noted earlier, communication with others allows one to access some of their personal knowledge. Thus, improved telecommunication systems make it easier to access the accumulated personal knowledge of others.

Computers and Public Knowledge

Computers give us a new way to collect, store, transmit and access information. Computers are at the heart of our rapidly expanding telecommunications system. It is estimated that by the year 1990 there will be one billion interconnected telephones on this planet (about one for every five people). Satellites and fiber optics are steadily decreasing the cost of long distance phone calls. There is a rapidly growing number of computerized databanks that can be accessed through the use of computers and our telephone system.

I assume you are all familiar with computerized databases and common uses of computers in storing and retrieving information. I never cease to be amazed at some of the systems that are now in routine use. For example, consider an airline reservation system. One can telephone a travel agency and schedule a flight in a matter of minutes. There are tens of thousands of computer terminals that access databases containing information about many millions of seats on tens of thousands of flights that are scheduled in the months ahead. The immense ticket sales problem can be satisfactorily solved only with the use of a high quality computer and telecommunications system.

One of the newer media for the storage and retrieval of information is the videodisc. One type of videodisc can store half an hour of television, with two sound tracks, on one side of one disc. Under computer control this medium can be used to provide interactive computer assisted learning. One videodisc can store 54,000 color pictures—for example, pictures of artwork, or pictures taken through a microscope for use in a biology class.

Perhaps you have heard of CD ROM (Compact Disc-Read Only Memory). A laser disc that is 4.72 inches (14 cm) in diameter and the thickness of a phonograph record stores about 550 million characters. This is roughly the equivalent of 500 very thick novels. A set of encyclopedias occupies only a modest fraction of one disc. A handful of these discs can contain the equivalent of an elementary or secondary school library. Under computer control, information in CD ROM databases can be rapidly retrieved.

In early March 1986, Microsoft held a major CD ROM conference in Seattle, Washington. While the main focus was the CD ROM, in some sense the show was stolen by a somewhat premature announcement of a new product for the home market: a computerized system that plays laser music disks and can read and process data (including computer programs) stored on CD ROMs. This system will cost perhaps $200 more than a system that can only play the laser music disks, and it may reenergize the home computer market.

This is not intended as a workshop on computerized storage and retrieval of information. Thus, we will close this aspect of computer impact on problem solving with a little brainstorming on possible applications and impacts.

Exercise. Suppose every student could have easy access to a computerized information retrieval system with a database equivalent to tens of thousands of books and an equivalent number of periodicals. How might this affect problem solving? How might education be changed to better prepare students for life in a society offering increasingly easy access to information?

Debriefing can be done in triads or in the whole group. It surprises me how much difficulty workshop participants have in coming up with good ideas during this exercise. Librarians and library media specialists tend to do the best. They seem to be more fully accepting of the idea that a standard way to approach the solving of any problem is to use a library to retrieve information on the problem. They have the library research skills that facilitate such an approach to problem solving.

Math teachers tend to have difficulty in this activity. Few math teachers emphasize the idea of retrieving information, except from one's own head or the course textbook. This is somewhat surprising, since math has a long and colorful history, and mathematicians are trained in the basic idea of building on the work of previous mathematicians.

Math teachers tend not to think of a handheld calculator as a device that both contains (is a storage medium for) computational algorithms and can execute the algorithms it contains. We will talk about calculators more in the next section.

Computers and Artifactual Knowledge

An electronic digital computer system is an artifact. It is a rather general-purpose aid to storing, processing, and retrieving information. Thus, it can directly aid people who need to process information, and it can be built into other machines to improve their capabilities. Computers have made possible the development of intelligent machines.

Let's examine handheld calculators, since they are an excellent example of artifactual intelligence. An inexpensive four-function calculator has stored within its circuitry algorithms to do addition, subtraction, multiplication and division of decimal numbers. Many humans have stored within their heads algorithms to perform the same operations, sometimes by making use of pencil and paper.

It takes hundreds of hours of study and practice for a typical person to memorize the basic computational algorithms and to develop reasonable speed and accuracy in applying these algorithms. Moreover, the human mind seems to be error prone in carrying out such detailed work. Thus, we are satisfied if a typical student can perform at the 80 or 90 percent accuracy level. This is adequate for passing required competency tests, but not particularly useful when working with real-world problems, where errors may have serious consequences. Would you like to fly in an airplane designed or engineered by people who made one error in every 10 computations required in the work?

Exercise. Think about the effects of providing all fourth grade students with handheld, solar-powered calculators and permission to use them whenever they please. This would be accompanied by a change in mathematics instruction from (mainly) computation to (mainly) problem solving. (Quite adequate calculators of this sort retail for about $5 and are apt to stand up to many years of use.)

Debrief in triads. For many people, this suggested use of calculators is an emotional, values-laden issue. Share your feelings on this type of calculator use.

A typical fourth grade student with a calculator and a couple of hours of practice can out-perform a (non-calculator-equipped) seventh grader in a computation test. That is, an inexpensive example of artifactual knowledge has the potential to make a large change in our mathematics education system.

I find that in a typical workshop, several participants get quite angry if I suggest that beginning in the fourth grade all students be given calculators and be allowed to use them whenever they like. On the average, teachers are cautiously supportive of this move. Most mathematics education leaders are strong supporters.

Every artifact embodies some of the knowledge and skills of the developer of the artifact. The idea of learning to use artifacts is as ancient as the history of humans. But computers have added a new dimension. A calculator is just the tip of the artificial intelligence iceberg. Computers provide a new aid to capturing human knowledge and skills in a form that can be used by others. This is such an important idea that we devote the entire next part of the workshop to it.

Exercise. Think of a problem that you have recently encountered and solved. To what extent did you make use of personal knowledge? To what extent did you make use of public knowledge? To what extent did you make use of artifactual knowledge?

Debrief in triads. To the extent that you are willing, share your responses with the members of your triad.

The personal knowledge versus public and artifactual knowledge issue ties in closely with the High Tech/High Touch Leadership Development Workshops that I give. The organization of our society and the current emphasis on technology provides more and more accumulated public knowledge and artifactual knowledge.

But personal knowledge is little impacted, and it remains the very essence of our being and our functioning as human beings. This type of analysis offers a hint as to one long-term direction schools might take. The role of teachers may increasingly be to foster understanding of personal knowledge and development of one's potential as a human being. Computer assisted instruction will take over more of the factually oriented instruction as well as many classroom management and record keeping details.

Applications

1. This application works well at the fourth grade or higher level. Select a worksheet of multiplication and division problems involving whole numbers and decimals. Get two volunteers. One is allowed to use a calculator while the other is not. Have a contest, and determine the results. (You may want to repeat the contest a couple of times, using different participants each time.) Then have the whole class discuss the results. You may find that your students consider it cheating to use a calculator.

   This application can lead to a class discussion of computational skill versus higher-order mathematical skills. What is an appropriate balance, and why? Seek the opinions of your students.

2. Have each student think of some tool, machine, or other artifact they frequently use. Each is to clearly specify what problem this helps him or her solve. To the extent possible, have them make lists of alternative ways of solving the problem.

   This same idea can be used in a class discussion. Have a student name a tool, machine, or other type of artifact. Then have the class suggest types of problems that are solved using it. Finally, brainstorm on other ways the problem could be solved without use of the artifact.

   This application is intended to increase students' awareness of how they routinely use artifacts to aid them in solving problems.

3. How much knowledge is stored in your school library? This is a profound question that can be used with students at almost any grade level. For example, if there are two copies of a particular book, is this twice as much knowledge as a single copy? Or, if two different books address nearly the same topic, how does one count them? Is there a difference between knowledge stored on a magnetic tape and knowledge stored in a book? How does one count tapes or records containing music?

The purpose of an application such as this is to get students to think about the accumulated knowledge stored in a library. How much is there, and what does it take to learn to make use of this stored knowledge?

4. If your students know how to use a computer, give them some introductory instruction in use of a graphics package. (Some graphics software for drawing bar, line and circle graphs is suitable for use even in the primary grades.) Then assign each student two graphics-oriented problems of nearly equal difficulty. One is to be done by hand and the other is to be done using the graphics package. Each student is to write a brief report (or participate in a class discussion) on these two different approaches to solving a graphics-oriented problem.

Activities

1. Do you make regular use of a calculator? Why, or why not? Express your personal philosophy about allowing students to make use of calculators as an aid to problem solving.
2. Consider a car as an aid to problem solving. Compare and contrast it with a computer as an aid to problem solving. What sorts of ideas are suggested by this analogy?
3. Consider the use of a word processor with spelling checker and grammar checker as an aid to solving a problem of needing to write a critical essay. What are your feelings and opinions as to whether it is all right for a student to make use of such aids to solving a writing problem? Compare/contrast your feelings and arguments with the use of a calculator or computer to solve a mathematics problem.
4. Do you pay a person to prepare your income tax forms? One way to solve a problem is to hire someone to solve it for you. In essence, this is a way you can make use of the personal knowledge of someone else. Make a list of problems that you solve in this way. Discuss the educational implications of this approach to problem solving.
5. In 1986 there were about 30,000 commercially available pieces of microcomputer software. Imagine the possibility of having access to a modem-equipped microcomputer system that has the following features:
    a. It can access a detailed index to and description of all 30,000 pieces of microcomputer software.
    b. It can access each piece of software and download it to your microcomputer so you can use it on your microcomputer. Each piece of software includes CAL materials designed to teach one how to use the program.

    Discuss how such a system might affect education and what constitutes appropriate training for students who would have easy access to such a system.

Part 5: Effective Procedures

What is an Effective Procedure?

When you are able to solve a particular type of problem routinely or automatically, you have developed one or more procedures (algorithms, detailed sets of directions, recipes) for this type of problem. Computer scientists are deeply concerned with developing procedures that tell a computer how to solve a certain category of problem. We will use the phrase effective procedure in discussing these ideas.

An effective procedure is a detailed, step-by-step set of instructions having the following two characteristics:

1. It is designed to solve a certain specified category of problems or a specific problem.
2. It can be mechanically interpreted and carried out by a specified agent. Here the term mechanical means in a machine-like, non thinking manner. Computer scientists are interested in situations where the agent is a computer or a computerized machine, such as a robot.

Of course, the agent in an effective procedure need not be a computer. Watch as I tie my shoe. (In the workshop, I give a demonstration of tying my shoe while continuing to lecture.) I am able to tie my shoe while at the same time carrying on a conversation. I have stored in my muscles and subconscious an effective procedure for shoe tying. Once I start executing the procedure, it proceeds while I use my conscious brain to carry on a conversation.

Exercise. To make sure you all understand the idea of an effective procedure, I'd like to hear a number of examples from you. When you give me an example, please indicate the problem being solved and the agent executing the instructions in the procedure.

Debrief. Workshop participants seem to have little trouble making an extensive list of effective procedures. Many are outside the realm of computer science. Some are borderline. For example, automation of a factory can make use of computers, but it doesn't in many cases.

Earlier in the workshop we briefly discussed the idea of chunking knowledge stored in one's brain. In essence, a procedure which one has memorized and routinized is a chunk of stored knowledge. Some procedures, such as how to ride a bicycle, remain available for use even if they are not practiced for many years. Other procedures, such as how to add a list of fractions or solve a quadratic equation, may gradually fade away with lack of use.

Proven Effective Procedure (PEP)

The above definition of effective procedure includes no requirement that the procedure actually succeed in solving the specified problem. When I am tying my shoe I may end up with a knot, or the lace may break. A student's bug-ridden program, designed to solve a specified homework problem, satisfies the definition of effective procedure.

Thus, we are very interested in having effective procedures that have been proven to work. Mathematicians address the problem of proving that an algorithm accomplishes a specified mathematical task. Computer scientists address the problem of proving that a particular computer program actually solves a specified problem. The techniques of proof used in mathematics and computer science overlap. That is, there is a growing mathematically oriented science for attempting to prove that an effective procedure actually solves its designated category of problems.

We will use the phrase Proven Effective Procedure (PEP) to designate an effective procedure that has been proven to solve the category of problems it addresses. There is a large and growing number of PEPs. Some involve use of computers and others do not.

The question sometimes arises as to what constitutes a proof. In mathematics and logic one starts with definitions and basic assumptions and then develops careful, logic-based proofs from these starting points. In essence, such proofs can only be made in mathematics and logic situations. (Actually, one might define the discipline of mathematics and logic so that it consists of those areas in which such proofs can be made.)

Computer scientists have made some progress in being able to give rigorous, mathematical proofs of the correctness of programs. But the process of proving the correctness of even a short program is quite difficult and time consuming. Thus, relatively few of the programs that are routinely used have actually been rigorously proven to be correct.

However, there is a large number of computer programs that are based on mathematical algorithms and a careful, logical analysis of a problem. That is, the program is based on underlying theory and mathematics that can be clearly stated and is generally accepted as correct by knowledgeable scientists familiar with the subject. These programs are carefully written, carefully tested, and used over a period of time. Even though they have not been mathematically proven to be correct, we would tend to include them as PEPs. Examples include most of the statistical and mathematical program library available on mainframe computer systems. Many programs designed to solve problems in the physical sciences fall into this same category.

Some of the programs in statistical, mathematical, and physical science program libraries are so complex that they will never be fully tested and fully debugged. They perform correctly over a wide range of problems. We tend to assume that these programs are PEPs and to use them as if they were.

Exercise. The idea of a PEP is fundamental to making use of previous work of others when solving problems. I would like each of you to think of one or more PEPs.

Debrief in triads and whole group. Share your examples with members of your triad. In the discussion, you will want to ask what constitutes a proof to you. Mathematicians understand the idea of proof as being rooted in careful and rigorous definitions, axioms, and chains of logic. This is possible in mathematics, since to a large extent mathematics is a system created by humans specifically for the purpose of allowing such rigor.

In physics we know certain things, such as Newton's laws of motion. When these laws of motion were initially discovered, they were subjected to careful study by some of the greatest thinkers of the time. They were used to make predictions, such as forecasting when an eclipse would occur. The use of Newton's laws of motion allowed the development of effective procedures to solve a wide variety of physics problems. For many years these were considered to be PEPs.

Actually, Newton's laws of motion ignore relativistic effects, and they are found to be increasingly inaccurate as the objects in question have increased velocity. This example suggests that the idea of a PEP is both deep and difficult.

Some disciplines seem to have more PEPs than others. Thus, workshop participants tend to suggest PEPs from mathematics and the physical sciences. They tend to offer few examples from the arts and humanities.

The computational algorithms students learn for the four basic operations are all PEPs. Or, are they? A person can easily make a mistake in carrying out the steps in a computational algorithm. I have seen computers that make errors in carrying out certain computations. This suggests that we may need to distinguish between something that theoretically is a PEP, and the real-world implementation of the supposed PEP. One can have an algorithm that has been mathematically proven to solve a certain category of problems. One can use great care in writing a computer program to implement the algorithm. Still, the task remains to prove that the computer program is actually a correct implementation of the algorithm. This can be quite difficult, since the computer program may be both long and complex. And even then, how do we know that the computer always functions perfectly? Might a slight wearing of a part cause a computer to make a one-bit error (changing a 1 into a 0 or vice versa) once every 100 billion operations?

My conclusion is that a computer PEP is one for which we have a very high level of confidence, even though we cannot have 100 percent confidence.

Heuristic Effective Procedure (HEP)

There is a good chance that a number of the examples suggested as possible PEPs make use of heuristics (rules of thumb, procedures that don't always work, and procedures that are not rooted in fundamental theory that allows proofs to be developed). In the card game of bridge, for example, many heuristics are available to the players. One heuristic is that in leading against a no trump contract, you should select your longest suit and lead your fourth highest card in that suit.

In poker, you may have heard the heuristic "Never draw to an inside straight." In taking a true-false test, if you have to make a guess on a question, a possible heuristic is to always guess true. This is because many teachers find it easier to construct true statements for use on tests. Weather forecasts are based on a combination of science and heuristics.

Exercise. Think about some heuristics you use in your everyday life. Share them with the group. In sharing a heuristic, indicate what problem is being addressed and how well the heuristic seems to work for you. Are you able to detect when the heuristic fails you?

Debrief. We all use heuristics all of the time. We do this because they work pretty well for us, and they are a good way to build on previous work of ourselves and others.

For example, when I want to shave in the morning I begin by plugging in my electric razor and turning it on. This beginning usually works. But every couple of years it fails because my razor has worn out.

This illustrates a very important idea. When I use personal heuristics, I can usually tell when they don't work. I am intimately involved with the problem and I have knowledge that helps me determine whether the procedure I am using is helping to solve the problem.

Contrast this with more complex heuristics, such as those used by our federal government in making economic decisions. There it is difficult, if not impossible, to determine if correct actions are being taken.

Paralleling the idea of a PEP, we have the idea of a Heuristic Effective Procedure (HEP). It is a quite important idea, especially if one insists on a very rigorous definition of proof. If one uses a sufficiently rigorous definition of proof, then essentially the only PEPs are those that are rooted in formal logic and mathematics, and we mainly use HEPs as we function in our everyday lives.

PEPs and HEPs play a central role in problem solving. Consider the following five-step model for handling a problem situation which you would like to resolve.

1. Work with the problem situation until you have converted it into a well-defined problem–that is, until you have identified and understood the Givens, Goal and Guidelines. This is a creative, higher-order thinking process, often involving considerable knowledge as well as a good sense of values.
2. Select and/or develop a PEP or HEP that is designed to solve the problem. This is an information retrieval and/or creative thinking step. (Usually it involves both; computers may be useful in retrieving needed information.)
3. Execute or cause to be executed the steps of the PEP or HEP. This may be a mechanical, non thinking step where speed and accuracy are often desired and computers may be quite useful.
4. Examine the results produced in step 3, working to determine whether the problem you defined in step 1 has been solved. If it has been solved, go to step 5. Otherwise, do one of the following:
   a. Return to step 2 and determine some other approach to solving the problem.
   b. Return to step 1 and determine some other problem to be solved.
   c. Give up.
5. Examine the results produced in step 3 to determine whether the original problem situation has been satisfactorily resolved. If it has, you are done. If it hasn't been satisfactorily resolved, do one of the following:
   a. Go to step 1 and determine some other problem to be solved.
   b. Give up.

Triad Group Exercise. Select a problem situation that you are willing to share in your triad. Test it against the five-step model, mentally contemplating carrying out each of the steps. Then "think out loud" for your triad members as you simulate in your mind actually carrying out the five steps.

Triad Group Exercise. Discuss what constitutes a good education for improving one's ability to carry out the five-step process for resolving problem situations. Pay particular attention to how the ideas of your triad compare with our present educational system and/or your own teaching style.

Debrief. This five-step model for handling problem situations can be used regardless of whether computers and computerized information retrieval systems are available. But their presence has a major impact, and the impact is strongest in the sciences and technology. The more science-like or math-like a problem is, the better chance that one can make use of previous work of oneself and others. Similarly, the more science-like or math-like a procedure is, the greater the likelihood that it can be executed by a computer.

I think we all realize that the tools we use, our aids to problem solving, influence our thought processes. If we want students to make effective use of computer related technology to resolve problem situations, we need to give students substantial training and experience in using the tools. Since there is a fixed amount of school time available, this means we need to reduce the time spent on some other topics and subjects.

Most people don't think much about similarities or differences between PEPs and HEPs. But computers are often used to execute both, and most people tend to associate computers with mathematics, rigor, certainty, etc. Thus, there is a tendency to believe that if a computer "says" something is so, it must be so. That is, most people equate HEPs with PEPs when the agent is a computer. That can be a rather large mistake, and making it frequently leads to serious errors in problem solving.

Exercise. Consider a computer-based medical diagnostic HEP that solves a certain category of diagnostic problems with 95 percent accuracy. How would you feel about this system being used to diagnose and recommend treatment for a medical problem you were having?

Or, consider an economic model that in the past was able to forecast economic developments six months into the future with an average error of less than five percent. If you were a business or government leader, would you base major decisions on such a computer model?

Debrief in triads and then in whole group. The purpose of the exercise is to increase awareness of the uncertainties inherent to solving real-world problems. Computerizing a solution procedure may increase the likelihood of success, but it doesn't guarantee certainty. It could be that the typical doctor is only 90 percent accurate in diagnosing the medical problem, so that one has a potential gain through use of the computer. But it could also be that the human doctor might detect a totally different medical problem while going through the process of attacking the presented problem.

Most people seem to have an intuitive sense about the difference between PEPs and HEPs. It might be expressed by a statement such as "Nothing in this world is certain." With a little training and experience, students also learn this about computers. This should be an important component of computer literacy instruction.

Practical, Computerizable PEPs and HEPs

Computer and information science focuses on PEPs and HEPs in which the specified agent is a computer. Some computer scientists have little concern about how long it might take an actual computer to solve a particular problem, while others pay particular attention to this issue.

For example, consider the game of chess. It has long been known how to write a computer program that could play a perfect game of chess. In essence this involves writing a program that would consider every possible move you might make, every possible response available to your opponent, every possible move you might make in response to your opponent's move, and so on. The trouble is, there are about 10^{120} possible sequences of moves. The fastest computers currently in existence could not examine this many possibilities in an amount of time equal to a trillion trillion trillion trillion trillion trillion times the age of the universe.

Likely you have easy access to a microcomputer such as those most frequently used in schools. These are powerful machines. They are more capable than some of the computers built in the 1960s that cost hundreds of thousands of dollars. But the power of a personal microcomputer shrinks to insignificance when compared to the multimillion dollar mainframe supercomputers currently in use. Consider, for example, the problem of making a reasonably accurate long-range weather forecast. It takes many hours of time on a supercomputer to do this. The same computations would take hundreds of years on an inexpensive microcomputer.

Such considerations lead to us introducing the phrases Practical Computerizable PEP (PC-PEP) and Practical Computerizable HEP (PC-HEP). These are PEPs and HEPs in which the agent is a computer. They have the added feature that the computer hardware and software available can execute the instructions at a reasonable cost and in a reasonable amount of time. We leave as subjective what constitutes reasonable (practical) cost and time.

The number of PC-PEPs and PC-HEPs is growing quite rapidly for three reasons:

1. Computer hardware is getting more powerful and less expensive.
2. Computer scientists and others are developing new procedures for attacking problems that have not yet been solved. Computer scientists are developing more efficient procedures for solving important categories of problems. These procedures require less compute power to accomplish their designated tasks.
3. Researchers in every discipline are advancing the frontiers of their disciplines, and quite a bit of their progress can be (and is being) represented in the form of computer programs.

Triad Group Exercise. In your triads, brainstorm a list of problems or types of problems for which currently available PC-PEPs and PC-HEPs cannot solve the problem or are not a major aid. After a list has been created, discuss which problems are apt to be removed by computer and other progress over the next few decades.

Debrief. It is hard to give examples of problems in which current PC-PEPs and PC-HEPs are not of value. But in some areas (such as music and psychotherapy) their use is peripheral, while in other areas (science and engineering) their use is central. The idea of artificially intelligent expert systems generally comes up in this discussion.

Expert Systems

One subfield of computer and information science is called artificial intelligence. Researchers in artificial intelligence have made slow but steady progress over the past 40 years. In recent years this progress has led to a number of marketable products as well as considerable publicity. The Japanese Fifth Generation project provides a good example of the publicity (Feigenbaum and McCorduck, 1984).

A knowledge-based expert system is a computer system designed to capture and make use of some of the knowledge a human expert uses in solving a certain category of problem. It is a PC-HEP. It is a unique, new combination of public knowledge and artifactual knowledge. Currently (1987) a few thousand PC-HEP expert systems are in everyday use, and there are many thousands more under development. This is a rapidly expanding application of computers.

Increasingly such expert systems will prove to be a challenge to our educational system. The key question is, "If a computer can solve or help solve a particular category of problem, what do we want students to learn about solving that category of problem?" Perhaps the largest issue here is the detection of errors. Can we help students to develop an intuitive sense or sufficient basic understanding of the concepts of the problem being solved so they remain in control and are the guiding force in the overall problem solving process?

Exercise. Think of a small, self-contained part of a discipline that you know very well—that is, in which you have a good level of expertise. (It might be some aspect of teaching.) Then think about what you know and can actually do using your knowledge of this area. To what extent do you think it would be possible to computerize this knowledge?

Debrief in triads. Share your example. The exercise illustrates part of the process involved in developing a knowledge-based expert system. One isolates a well-defined, limited task involving knowledge and problem solving skill in which one or more humans have great skill. One then works to identify and computerize the processes the humans use to solve problems in the area.

At the current time it is both time consuming and reasonably expensive to build a knowledge-based expert system, but the cost is rapidly declining. In some sense building such a system is still an art, rather than a science. Indeed, many people argue that it is a very long way from being a science, and that researchers in artificial intelligence have barely scratched the surface of this endeavor. But current researchers are building on the work of previous researchers, and cumulative progress is occurring.

Applications

The ideas of effective procedure discussed in this part of the workshop are some of the most important intellectual ideas of the 20th century. Your students will live their adult lives in a world that makes steadily increasing use of PC-PEPs and PC-HEPs.

One response educators have suggested to this is that students should learn about procedural thinking. That is, students should study procedures, learn to represent procedures, and learn to think about roles of procedures in problem solving. The applications below follow that suggestion.

1. Divide your students into pairs and have each student select a procedure involving physical activity, such as tying a shoe, shooting a basket, walking, etc. Each student is to attempt to communicate in words, without any body language, how to carry out the procedure. (You and a student may first want to role play this activity.)

   There are two purposes to the application. First, students are led to think carefully about a procedure—thereby increasing their knowledge about procedures and procedural thinking. Second, students will learn that certain procedures are quite difficult to communicate through use of the spoken (or written) word.

2. Have your students think about the problem of determining the oldest and youngest students in the class. If your students are very young, ask for a volunteer to tell how to select the oldest person in the class, and a second volunteer to tell how to select the youngest. With older students, the exercise can be to write down procedures for determining the oldest and youngest students in the class.

   The purpose of this application is to give students practice in developing a procedure that can be proven to work and that can be computerized. Help your students realize that the problem can be solved by hand or by computer, and lead a discussion on the advantages and disadvantages of each approach.

3. This exercise can be adjusted to fit various grade levels and can be done orally or in writing. Select two locations that your students know about. (For very young students, this might be two places at school. For older students, it might be two different places in town, or still further apart.) The exercise is to give precise instructions that someone else could follow in going from the first to the second place.

   Some people seem to be much better than others at giving directions. Some are better than others at being able to receive and follow directions. A map is a useful aid in solving the problem of giving directions. Notice that if one has learned to read a map this skill transfers to a wide range of map reading situations.

4. Select a computer program that solves or helps solve a category of problems your students are learning to solve without the use of a computer. Have your students use this software. Then have your students write a report, or carry on a class discussion, about the two different approaches to problem solving.

   The idea is to address the issue of what one should learn to do mentally, assisted by books and other conventional aids to information retrieval, assisted by pencil and paper, and assisted by computers. Student opinions on this are interesting and important.

5. What are some things that a human can do but a computer cannot do? Have your students discuss or write about this question.

   This exercise can increase your students' awareness of humans competing with computers versus humans learning to work with computers. Perhaps education should focus more on developing the truly unique human potentials and place less emphasis on helping students learn to do things that computers can do.

6. If a computer is given a problem to solve and it produces an answer, how can one know it is a correct answer? Engage your class in a discussion of this issue. You might begin by asking the same question for a human solving a problem.

   The intent of this exercise is to increase student awareness of the importance and the difficulty of having methods for checking a proposed solution for correctness. The real world does not have a teacher's answer key!

   The issue raised in this exercise is one of the most important issues in the computer applications area. As more and more computer applications are developed and put in place, people in our society will become more and more dependent on computers. The computers will be producing answers to very complex problems. But most of the answers will be based on PC-HEPs. For many, there will be no easy way to tell whether the answers produced are correct, or even reasonably correct.

Activities

1. Select a discipline that you know well, and make a list of PEPs and HEPs that are useful in solving some of the problems of that discipline. Which of these seem like they could probably be computerized now, or have already been computerized?

Which seem like they will be very difficult to computerize or may never be computerized?

2. Select a colleague and/or some students, and explain the idea of PC-PEPs and PC-HEPs. Are you able to effectively communicate the key ideas? Write a brief report on this exercise.
3. Make a list of problem solving activities that you routinely accomplish, but for which you feel no PC-HEP or PC-PEP will exist in the next 20 years. Discuss the curriculum design implications of your list.
4. Consider a hypothetical situation in which every student and teacher in a school has a portable, very powerful computer and access to all existing educationally oriented software. (Current estimates are that there are about 10,000 pieces of educationally oriented software commercially available. The Educational Software Selector, published by EPIE in New York, lists about 8,000 pieces of educational software for microcomputers.) Moreover, assume that all students and teachers have had substantial instruction and experience in using the computer systems. Make an estimate of the average number of hours per week a student might use such a computer. Explain the basis of your estimate. Discuss what might happen over the next five years that could lead to an increase or decrease in the possible level of use, and give an estimate for usage levels five years from now.
5. The Strategic Defense Initiative (Star Wars) proposes developing a computerized system that could destroy enemy missiles. The necessary computer programs would be millions of instructions in length. They would process data from a number of sensing devices, such as radar systems. They would control laser and other weapons for destroying missiles. Some of the software would be PC-PEPs and other parts would be PC-HEPs.

   Some opponents of the Strategic Defense System argue that it would be impossible to write such a computer program and have reasonable confidence in its correct functioning. Discuss the merits of this type of argument based on your understanding of PC-PEPs, PC-HEPs and other relevant factors.

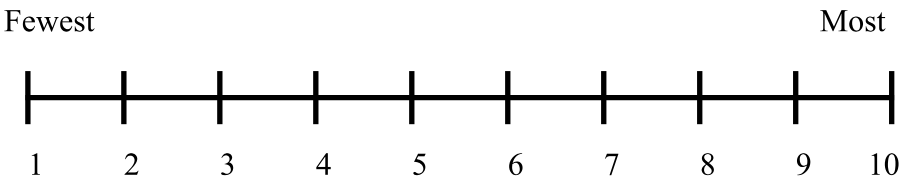
Part 6: Conclusions and Recommendations

PC-PEPs and PC-HEPs in Various Disciplines

If there were few PC-PEPs and PC-HEPs, then educators would not need to give much consideration to the idea they represent. But as we have seen, their numbers are large and rapidly growing.

In a country, state, school district, or school with few computers, the idea of PC-PEPs and PC-HEPs may seem largely irrelevant to school teachers and to the precollege education system. But one characteristic of the Information Era in which we now live is that the whole world is increasingly a single marketplace and there is increasing economic competition among countries, states, and local regions. This means that a country or state can gain economic advantage by providing its citizens with aids to thinking and problem solving, and teaching them to make effective use of these aids. Some have begun to do so. A high quality educational system is important to a country competing in international markets.

Exercise. Consider a PC-PEP/PC-HEP scale (see diagram) numbered from 1 to 10.



Discipline Scale of PC-PEPs and PC-HEPs

Select an academic discipline that you feel has the fewest PC-PEPs and PC-HEPs, and use it to define the lower end of the scale. Select another discipline that you feel has the most PC-PEPs and PC-HEPs, and call it a 10 on your scale. Finally, put the discipline of education (as it now exists) on your scale.

Debrief. If time permits, I first have workshop participants share and discuss their results in triads. Then I ask for examples of disciplines used to represent 1 and 10 on the scale. Typical 1 responses are art, poetry, writing, psychotherapy and political science. Notice that these tend to require good interpersonal skills. The arts and humanities all fall on the left half of the scale.

Most workshop participants use mathematics to define a 10, although a few suggest physics or chemistry. The sciences all receive high ratings on this scale.

I feel that one possible outcome of the increased computerization of our society is that our schools will begin to place increased emphasis on the arts and humanities. My logic is as follows. Currently our schools place considerable emphasis on students gaining good paper-and-pencil skills in carrying out a number of PEPs and HEPs. Increasingly, however, computers will be used to execute these PEPs and HEPs. Schools will help students learn what types of problems can be solved by use of PC-HEPs and PC-PEPs, but there will less value in students developing good skills in solving such problems by hand. For many students, some of the instructional time that is saved will be devoted to additional study of the arts and humanities, and in developing improved interpersonal skills.

On the ten-point scale, education generally gets about a 3 or 4 from most workshop participants. Surprisingly to me, however, is that it almost always receive a couple of 2s and a couple of 6s or 7s. If time permits, it is fun to have the 2s and 7s discuss their views of education.

I often ask workshop participants to indicate where they think education will be on the scale 20 years from now. Typically it moves up one point on the scale. Progress in learning theory, teaching theory and special education supports this movement. Twenty years from now our educational system will be making substantial use of CAL materials that have been carefully tested and are based upon sound theories of teaching and learning.

The Human-Machine Interface

The major theme of this booklet is that all students should learn quite a bit about roles of computers in problem solving. Many educators have supported this position since the mid 1970s, when the idea of computer literacy began to become widely accepted. In recent years there has been a bandwagon effect, so that some states now require all students to receive instruction designed to make them computer literate.

The issue is not whether students should learn to make use of computers in problem solving. Rather, the issue is how to accomplish this goal. In the early history of computers, it was felt that every computer user needed to know a great deal about computers. If you wanted to use a computer, you studied machine and/or assembler language. The goal was to help you gain an intimate understanding of the machine.

FORTRAN, developed from 1954 to 1957, represented a major breakthrough in improving the human-machine interface. For a person with a math-oriented, problem-solving background, Fortran could be learned quickly. Moreover, Fortran was machine independent. This meant that one didn't have to learn a new language each time a new computer was developed.

Other high-level languages were developed to meet the needs of specific groups of problem solvers. For example, COBOL was designed to aid in solving business problems, and BASIC was designed to fit the needs of college students. Each new programming language is designed to improve the human-machine interface for a particular group of computer users.

From very early on, however, there were many people who wanted to use computers but who didn't want to become skillful programmers. They merely wanted to build on the work of programmers—to use canned programs. Thus, the computer industry has devoted considerable effort to improving the human-machine interface for nonprogrammers. Even by the early 1960s, large libraries of computer programs had been developed. Using these, nonprogrammers could carry out sophisticated statistical calculations or solve quite complex applied mathematics problems.

Timeshared computing represented a major improvement in the human-machine interface. Microcomputers represent still another major improvement. In recent years, applications software has further improved the human-machine interface. The touch screen, the mouse, voice output and voice input, etc., all help to improve this interface.

The trend is obvious. The human-machine interface will continue to improve. This means that it will become easier and easier to learn the rudiments of using computers. To cite a single example, many libraries have replaced their card catalogs with computerized systems. A typical person requires only a few minutes of instruction and practice to gain a rudimentary but functional level of skill in the use of such a system.

This progress in improving the human-machine interface leads to the questions of who needs to learn to program and what can one learn about problem solving through studying computer programming. Gradually, many instructors in computer science courses have come to realize that it is problem solving—not programming—that is at the very heart of the courses they teach. Thus, such courses are placing increased emphasis on problem solving in a computer environment. These courses can provide an excellent environment for studying and practicing certain types of problem solving.

Some people argue that computer programming provides such an excellent environment for developing problem-solving skills that students gain a major benefit by learning to write programs. They conjecture that the problem solving skills needed to write computer programs readily transfer to solving non computer problems. Currently, however, the research literature provides relatively little support for this conjecture. That is, there have been quite a number of research studies that hoped to prove that there is a large transfer of general problem solving skills from a computer programming course to non computer-oriented problem solving. A few of these studies have produced a few small indications that such transfer occurs, but most have failed to produce any significant results. My conclusion is that little transfer occurs unless the instructor places a major emphasis on activities likely to increase such transfer.

Of course, there are other good arguments to support exposing all students to computer programming. A programming language is designed for the precise representation of an effective procedure. Thus, programming gives students instruction and practice in representing effective procedures. It gives practice in procedural thinking. It helps dispel the magic of computer capabilities and limitations. It is a good environment for coming to understand the idea of PC-PEPs and PC-HEPs.

Observations

It is evident that computers will affect problem solving in some disciplines much more than in others. It is also evident that our school systems have not yet begun to effectively address the computer as a general-purpose aid to problem solving.

There are a few exceptions to this. In higher education some schools now require all of their students to own a microcomputer, or they provide ample equipment so that all students can have easy access.

We now have a few "classrooms of tomorrow" and "schools of the future" at the precollege level. These are experimental situations in which all students have a great deal of computer access. In some of the experiments there are two microcomputers for every student--one for use at school and one for use at home. These experiments are beginning to yield valuable data about what might be gained by students in a computer-rich environment, and what might be lost.

A major issue that has arisen in these computer-rich experimental schools is how much to use computers to teach basic skills and how much to use them to work on higher-order skills. Preliminary research results indicate that if the emphasis is placed on computer-as-tool and on problem solving, students do not make greater-than-average test score gains on standardized tests. That is, it appears that the standardized tests used in national assessments are heavily weighted toward basic skills. If we want students to score higher on these tests, we can achieve this by lots of use of routine drill and practice software aimed at the basic skills. But research indicates that this does not improve higher-order skills.

My conclusion from these studies is that we should take advantage of the CAL designed to help students gain mastery of basic skills. Research indicates a substantial gain in rates of learning is achieved through use of such software. (Many studies show students learn 50 percent to 100 percent faster.) The time saved could then be used to place increased emphasis on higher-order skills. This approach, however, must be tempered by a good understanding of what basic skills are important to learn. I would eliminate at least 100 hours of instruction and practice in paper and pencil long division of multi-digit numbers from the curriculum, and replace it by allowing students to use calculators. I would eliminate a similar amount of instruction in performing the four basic arithmetic calculations on fractions.

The whole curriculum needs to be carefully examined in light of our knowledge of lower-order and higher-order skills, the increasing capability of calculators and computers, and the increasing availability of these aids to problem solving.

I feel that integration of the computer as an everyday tool throughout the curriculum is the most important and challenging task facing computer educators. (But once again, that is another workshop. It is interesting to see how problem solving can serve as a central theme in bring together all aspects of computers in education.)

Recommendations

This workshop has analyzed some generally applicable ideas about problem solving and ways in which computers affect problem solving. An underlying theme of the workshop has been a recommendation that problem solving be given increased emphasis throughout the curriculum. This can be done both by having students study general methods of problem solving and by having students spend a great deal more time actually solving problems. Some recommendations more specifically related to computers are given below.

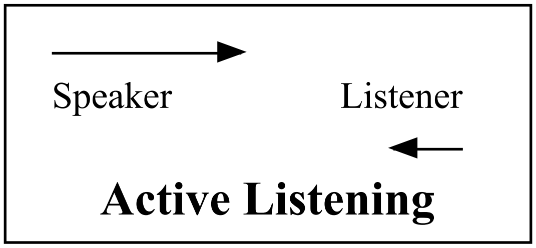
1. The idea of effective procedure (which includes PC-PEP and PC-HEP) is among the most important academic ideas of our century. I recommend that all students learn these ideas and their impact on various aspects of what it means to be educated for life in our society.
2. Accessing, organizing, processing, and storing information are central ideas in problem solving. Computers are very useful in carrying out these activities. I recommend that all students learn to make routine use of computers to access, organize, process, and store information. This use should occur in all courses in the curriculum.
3. Computers are a useful aid to actually solving problems in every academic discipline, and this importance is growing. I recommend that every course include information about how computers can help solve the problems being studied and how computers help create problems within the discipline the course is covering.
4. Computer assisted learning can help many students to better and more rapidly gain basic skills and knowledge essential to problem solving. I recommend increased use of CAL for these purposes.

5. Computer simulations can provide rich problem solving environments. I recommend that schools make increased use of computer simulations to give students practice and appropriate feedback in problem solving—especially in interdisciplinary problem solving.
6. Most real world problems are interdisciplinary in nature. I recommend that schools place increased emphasis on cross fertilization among disciplines, applications of one discipline to the study of a second, and solving problems making use of information and ideas from several disciplines. The computer can help motivate this change in educational emphasis, and it is a valuable tool in such problem solving.
7. Problem solving is at the very heart of computer science. I recommend that computer programming and computer science courses place increased emphasis on problem solving.
8. Good teaching can increase transfer of learning. The goal is to help students learn to transfer their problem solving knowledge, skills, and techniques to problems throughout the curriculum and to real world problems. I recommend that all teachers place increased emphasis on transfer of learning.
9. Educators have a professional responsibility to remain current in their disciplines. I recommend that all teachers become functionally computer literate in using computers as an aid to learning/teaching, using computers as an aid to problem solving within the specific subjects they teach, and using computers as a general-purpose aid to problem solving in our society.

The stated purpose of this workshop was to increase your knowledge of roles of computers in problem solving. But throughout the workshop we have emphasized using your increased knowledge to improve our educational system. Take a few moments now to review key ideas in your mind. What will you do differently as a consequence of being in this workshop? Get at least one idea firmly in mind. Leave here with a resolve to implement your idea.

Appendix A: Active Listening

Active listening is a communication skill that is useful to everybody, and it can be used in every conversational setting. The essence of active listening is to listen, to work hard to understand, and to sense/receive the underlying feelings and meaning inherent to the communication. The best active listeners are simultaneously concentrating and relaxing. Active listening recognizes that a significant part of communication is nonverbal, and that one's ability to read or perceive nonverbal communications improves with practice and training.



Note that most of the action is Speaker ---> Listener.

Brief Guide To Active Listening

1. Pay attention to the speaker. Maintain eye contact and observe body posture, gestures, breathing, tone of voice, and skin color. Be especially aware of changes and relate them (in your mind) to what is being communicated at the time. Often you will perceive differences between verbal language and nonverbal language—such as saying yes while at the same time shaking one's head no.
2. Provide feedback to show that you are listening and understand. This feedback might include things such as:
   - A. Continuing to do 1. above.
   - B. Nodding one's head appropriately, while murmuring encouraging sounds such as yes, okay, I understand, go on, etc.
   - C. Paraphrasing and/or restating brief summaries. It can be quite effective to make use of appropriate words from the speaker's vocabulary when providing these summaries. However, extensive paraphrasing or other types of repetition of the speaker's message tends to be distracting and inappropriate.
3. Ask questions only when you do not understand what the speaker means. By and large, do not ask leading questions and do not ask questions that can be answered by a yes or a no. Questions should be open ended, providing the speaker with options to proceed in a direction he or she selects. Active Listening is not a courtroom interrogation.
4. Seek the underlying meanings and underlying feelings being communicated. Feedback should reflect that you are receiving the underlying meaning and feelings.

This is hard work, requiring full use of one's senses. The ability to be a good active listener improves with practice.

5. Provide positive strokes to the speaker. Your listening and paying active attention is a positive stroke. Understanding the communication is a positive stroke. An honestly felt final comment such as "Thank you for sharing." is a positive stroke.

I often begin workshops with an active listening exercise. Typically I will ask for a volunteer to talk to me and then I'll illustrate active listening in the conversation later. I debrief the process that was being demonstrated in the conversation. Next, I have workshop participants read some material (given above) on active listening. I stress the importance of nonverbal communication and of giving positive strokes. My feeling is that most people (especially educators) don't get enough positive strokes. Finally, I have workshop participants carry out an exercise such as the one given below.

Exercise. Divide into triads (groups of three). Designate one person to be speaker, one to be listener, one to be observer. Speaker is to spend one to two minutes on "What I expect to get out of being here today." A slightly different topic is "What I need in order to be a more effective computer education leader." Either topic is appropriate, as are other similar topics. Listener uses active listening techniques. Observer observes and acts as timekeeper. After one to two minutes, observer provides feedback to the pair for one to two minutes. Then switch roles; everybody should practice all three roles.

Debrief. I am surprised at how quickly and how deeply workshop participants get involved in the active listening exercise. It is almost as though they are starved for the opportunity to carry on a deep conversation about a topic of personal interest. Actually, the type of sharing being done in the triads is relatively rare in professional, academic circles. We tend not to know what our colleagues are feeling and thinking.

The observer has the opportunity to learn what seems to work and what seems not to work in active listening. Some observers report difficulty in not entering the conversation. For myself, I find that I learn more when I am the observer than when I am the speaker or listener. In any event, for all concerned, this is a useful learning experience.

In the debriefing, I always emphasize the importance of nonverbal communication. I have read and heard estimates suggesting that in face-to-face communication, perhaps two-thirds or more of the information is communicated nonverbally. My own experience in communicating using electronic mail or bulletin board systems certainly supports such estimates.

Appendix B: Thoughts on Computer Programming

Note to reader. Appendix B was not part of the original book. It was written for the "next" revision and printing, but such a revision never occurred. It is included here in this 2004 reprint for historical purposes.

The Computers and Problem Solving workshop is intended for a wide range of educators and does not have any specific computer prerequisite. Thus, there is no deep discussion of computer programming during the workshop. That is, no computer programming code is examined nor are any algorithms analyzed. There are no detailed comparisons of the relative merits of structured and nonstructured programming languages. There is no discussion about how to teach computer programming and how to emphasize problem solving while teaching computer programming.

Of course, all of these are important topics and all are suitable for inclusion in a Computers and Problem Solving workshop. Therefore, I have decided to include a few of my thoughts on computer programming in this appendix. As with the rest of the book, however, the technical level of this Appendix is intended to be quite low.

Computer Programming Versus Computer Science

I first used a computer in 1959. I wrote my first computer programs in 1960, and I have been quite involved in the computer field since then. When I first became involved with computers, there were very few computer science departments and very few computer-oriented courses. The courses that were available were designed mainly for people who were skilled at problem solving within their own professional fields.

Courses in FORTRAN programming illustrate what I mean. FORTRAN was developed during 1954-1957 (that is, it began to be available roughly in 1957) mainly to fit the needs of practicing scientists and engineers. The typical person taking a FORTRAN course had a bachelor's or master's degree in a science or engineering field and a solid background in mathematics. Thus, the course needed only to teach the rudiments of computer hardware and the programming language. People in the course had little difficulty applying what they learned to problems within their own fields. They already knew how to solve the problems within their fields, using calculators and manual methods.

In these early courses, essentially no class time was spent on teaching problem solving. Some time was spent on techniques from numerical analysis (a branch of mathematics focusing on developing algorithms for the computation of solutions to a wide variety of applied math problems). Occasionally class time would be spent developing and exploring an algorithm for ordering a list of numbers from largest to smallest, or alphabetizing a list of names. That was the extent of the computer science in many of these courses.

Gradually computers became more available and less expensive to use. Courses in computer programming came into the undergraduate curriculum. We began to get students into these programming courses who had relatively little formal training in solving math-oriented or science-oriented problems. One result was that many students did poorly in the courses. They could learn the rudiments of the programming language and computer system, but they had great difficulty in figuring out how to actually solve the assigned homework problems.

One of the first books I wrote was on flowcharting, and it was specifically designed for use in programming courses for teachers. The book was titled *How Computers Do It* and was published by Wadsworth Publishing Company in 1969. I reasoned that people who were having trouble on the algorithm development and problem solving components of a programming course could benefit by focusing on flow charting as an aid to algorithm development and problem solving. That is, I attempted to separate problem solving from learning the details of a programming language.

By the early 1970s, computer programming courses abounded in four- year colleges and community colleges. Many high school students learned to program. Some of the young students in these courses exhibited a remarkable capacity to learn about computers and computer programming, and then to apply this knowledge. A few went on to develop marketable pieces of software, to start computer companies, or perhaps to dabble in a life of computer crime, etc. In essence, these successful young students had a certain type of skill in problem solving, plus tenacity and stamina, that all came together in a computer environment. They were sometimes called computer hackers.

Most young students in these computer programming courses did not achieve such immediate success. Gradually, faculty in these courses came to realize that the young students (on average) were weak in problem solving skills. No matter how well the faculty taught computer programming, most of the students were relatively poor at applying their knowledge to solving complex, real world problems. (How can a student be expected to write an accounting program when the student doesn't know a debit from a credit and has never even had an introductory accounting course?)

In many colleges and universities the faculty teaching computer programming came to realize that their courses needed two additional focuses. They needed to begin to build a solid foundation of computer science, and they needed to place much greater emphasis on problem solving. The modern Introduction to Computer Science course is a careful blend among the three topics: computer programming, computer science (underlying theory of computer programming and computers), and problem solving in a computer environment. Usually the college-level version of this course has three or four years of high school mathematics (beginning with first-year high school algebra) as prerequisite. The course is often quite math oriented. Indeed, computer science majors generally take about three full years of college math in the better computer science departments.

Precollege Computer Programming

A few precollege students were introduced to computer programming during the 1950s. But it wasn't until the development of timeshared computing and timeshared BASIC in the mid to late 1960s that a significant number of these students got a chance to learn computer programming. Beginning in the late 1970s, microcomputers accelerated the trend toward teaching computer programming to precollege students.

It was soon discovered that the rudiments of computer programming can be taught even to very young students. With appropriate instruction, primary school students can learn a few of the statements in a programming language and learn to write short programs. This was initially demonstrated with BASIC and then more fully demonstrated with Logo.

People developing these courses ran into the difficulty of what problems to have students solve. Much of the power of BASIC, for example, is in solving math problems in an environment of algebra and calculus. Given below are a couple of sample programs that are somewhat typical of those used in elementary school.

```
10 PRINT "WHAT IS YOUR NAME?"
20 INPUT N$
30 PRINT "HELLO ";N$
40 END
```

```
10 PRINT "THIS PROGRAM ADDS TWO NUMBERS."
20 PRINT "PLEASE TYPE IN THE FIRST NUMBER."
30 INPUT A
40 PRINT "PLEASE TYPE IN THE SECOND NUMBER."
50 INPUT B
60 PRINT "THE SUM OF THE TWO NUMBERS IS ";A+B
70 END
```

I often wonder what students learn from writing and/or using such programs. Neither one involves a deep problem, so I suspect that little learning about problem solving occurs. Each involves variables, which are a profound concept in mathematics. Moreover, one program involves numerical variables and the other involves a character string variable. In combination, a deep idea is being introduced and illustrated. Both programs are (linear) step-by-step sets of directions that can be mechanically interpreted and carried out by a specified agent. Thus, they are effective procedures. Each program interacts with the program user, illustrating human-to-machine interfaces.

The previous paragraph illustrates a major challenge in introducing computer programming into elementary school. If the teacher is sufficiently knowledgeable, the environment allows presentation and discussion of the ideas of variable, effective procedure, and human-machine interface. Since these are very important ideas, the learning experience can be very valuable. But if the teacher is merely teaching programming, this excellent opportunity to acquaint students with variable, effective procedure and human-machine interface may be wasted.

Effective Procedure

In this section, the term effective procedure is used to mean a step-by-step set of directions that can be mechanically interpreted and carried out by a specified agent. In simpler terms, it is a computer program (thinking of the specified agent as a computer.)

At one time a number of people argued that many students would achieve high paying computer programming jobs immediately out of high school if they could have the opportunity to study programming in high school. But in just a few years this job market was flooded by community college graduates who had far more computer training than high school students could achieve. Now we are about to have a glut of four-year computer science graduates. Thus, people no longer argue the value of precollege computer programming as preparation for a job upon graduation from high school or a community college.

Several good arguments for teaching computer programming at the precollege level still remain. For example, one can argue the merits of exposing science-oriented, college-bound students to computer programming, suggesting that this gives them a distinct advantage in the computer science or computer programming courses they will face as freshmen in college. One can argue that some students have tremendous potential in this field and that early discovery of the talent may have a profound impact on their lives.

But I believe the best arguments are based on problem solving and the idea of effective procedure. Computers can be used to help create an excellent environment for learning and practicing a variety of important problem solving ideas. And, since the idea of effective procedure originated in computer science, one can argue that it is best taught in a computer environment.

Let me give an example. Humans have developed written symbols that can be used in the representation of problems. It is difficult to appreciate the power of representing a problem using the symbols and language of chemistry, physics, mathematics, or music. There is a substantial amount of knowledge built into the representational system of a discipline. Difficult problems can become simple if appropriately represented. Moreover, such representation itself is a great aid to accumulating knowledge in a field and building on previous work of others.

Thus, one of the most important ideas in problem solving is a combination of learning a problem representational system and learning to represent problems in the system. We can practice this in any discipline, since every discipline has its distinct vocabulary and representational systems. In particular, we can practice it in a computer programming environment.

Logo

Seymour Papert and the other people who invented and developed Logo had a goal of developing a programming language that would create a rich environment to practice representing and solving problems. Logo was designed to be relatively simple. Parts of it, at least, are quite easy to learn. The Logo environment was also designed to be stimulating and rewarding.

The designers and developers of Logo succeeded in their task. With just a very little learning of a computer system and some parts of Logo, students can accomplish programming tasks that they find exciting and rewarding. But over the years we have learned that that is not enough. Two key questions remain:

1. Does practice in representing and solving problems using Logo have a significant transfer to the same activity in any other discipline?
2. Do students learn the idea of effective procedure and its key role in problem solving through working in a Logo environment?

It would be nice if the clear and overwhelming answer to the questions were "Yes!" Unfortunately, there is little research to suggest this is the case. Moreover, resources devoted to creating a Logo environment and student time spent working in that environment could be used in other ways. Perhaps the introduction of inexpensive math manipulatives into the first three grades is far more effective in achieving the goals underlying 1 and 2 above. Perhaps an increased emphasis on science education in the upper elementary grades is a better way of achieving these goals. Perhaps teaching all students to read music, to play a musical instrument, and to compose for that instrument is more effective. The point is, we don't know!

My conclusion is that Logo does not create a teacher-proof environment in which important educational goals are automatically achieved. Logo can be used to help create an excellent learning environment with some unique features that cannot be created without computers. But the teacher remains a key part of the environment. If the teacher understands and builds upon important ideas about problem solving such as representation of problems, effective procedures, and transfer of learning, then the Logo experience becomes a valuable part of the curriculum. If the teacher lacks this preparation and knowledge, the resources might be better spent in areas where the teacher is better prepared.

Final Remarks

The rapid proliferation of microcomputers into our schools is bring with it powerful computer applications such as word processor, spread sheet, database, paint and draw graphics, and email. It is also bringing electronic encyclopedias and other resource materials. These types of applications are immediately useful to students, and they help the students to solve problems that are like those in the current curriculum. This greatly decreases the perceived value of having students learn to write computer programs.

However, the changes that are going on do not decrease the value of students learning about procedures, procedural thinking, and problem solving in environments that include computers. Problem solving remains as a core goal in education.

References

A Nation at Risk: The Imperative for Educational Reform. (1983). Washington, DC: National Commission for Excellence in Education, U.S. Government Printing Office.

Beyer, B.K. (March 1984). “Improving Thinking Skills: Defining the Problem.” Phi Delta Kappan, 486-490.

Beyer, B.K. (April 1984). “Improving Thinking Skills: Practical Approaches.” Phi Delta Kappan, 556-560.

de Bono, E. (1971). New Think. Avon Books.

de Bono, E. (1973). Lateral Thinking: Creativity Step by Step. Harper Colophan Books, Harper and Row.

ERIC. (December 1984). “Improving Students’ Thinking Skills.” The Best of ERIC. ERIC Clearinghouse on Educational Management, University of Oregon.

Feigenbaum, E.A., & McCorduck, P. (1984). The Fifth Generation: Artificial Intelligence and Japan’s Computer Challenge to the World. Signet (New American Library).

Fredericksen, N. (1984). “Implications of Cognitive Theory for Instruction in Problem Solving.” Review of Educational Research, 54 , 363-407.

Gardner, H. (1983). Frames of Mind: The Theory of Multiple Intelligences. Basic Books.

Jastrow, R. (1981). The Enchanted Loom. Simon and Schuster.

Kulik, J.A., Bangert, R.L., & Williams, G.W. (1983). “Effects of Computer-Based Teaching on Secondary School Students.” Journal of Educational Psychology, 75, 19-26.

Mayer, R.E. (1977). Thinking and Problem Solving: An Introduction to Human Cognition and Learning.

Moursund, D. (1986). High Tech/High Touch: A Computer Education Leadership Development Workshop, Third Edition. International Council for Computers in Education.

Naisbitt, J. (1984). Megatrends: Ten New Directions Transforming Our Lives. Warner Books.

Papert, S. (1980). Mindstorms: Children, Computers and Powerful Ideas. Basic Books.

Polya, G. (1957). How to Solve It: A New Aspect of Mathematical Method. Princeton University Press.

Rubinstein, M. F. (1975). Patterns of Problem Solving. Prentice-Hall.

Rubinstein, M. F. (1986). Tools for Thinking and Problem Solving. Prentice-Hall.

Toffler, A. (1980). The Third Wave. Bantam Books.

Torrance, J.P. Torrance is the author of a large amount of material on creative problem solving. For more information contact: Georgia Studies of Creative Behavior, Dept. of Educational Psychology, The University of Georgia, Athens, GA 30602.

Tuma, D, & Reif, F. (Eds.) (1980). Problem Solving and Education—Issues in Teaching and Research. Lawrence Erlbaum, Hillsdale, NJ.

Wickelgren, W.A. (1974). How to Solve Problems: Elements of a Theory of Problems and Problem Solving. W.H. Freeman and Company.