DOCUMENT RESUME

ED 302 217 IR 013 578

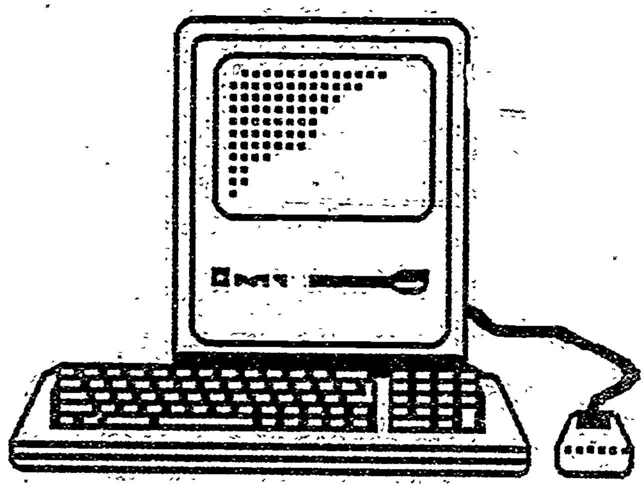
AUTHOR Moursund, David
TITLE Computers and Problem Solving: An Independent Study Course.
INSTITUTION International Council for Computers in Education, Eugene, Oreg.
REPORT NO ISBN-0-924667-40-0
PUB DATE 87
NOTE 230p.
AVAILABLE FROM International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403 ($18.00 prepaid. $2.50 must be added per book for all invoiced orders).
PUB TYPE Guides - Classroom Use - Materials (For Learner) (051)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Annotated Bibliographies; *Computer Assisted Instruction; Courseware; Curriculum Enrichment; Elementary Secondary Education; Futures (of Society); Higher Education; Independent Study; Inservice Teacher Education; Mathematics Instruction; Microcomputers; *Problem Solving; Textbooks

ABSTRACT

The first chapter of this text for an independent study course on computers and problem solving provides an overview of computers in education which focuses on the three main instructional uses of computers in schools--learning about computers, the computer-as-tool, and computer-assisted instruction. Using a workshop format, chapter 2 discusses lower-order versus higher-order skills and the possible role of computers in this area. Chapter 3 discusses the future of computers in education, including trends in hardware, software, networking, computer science, and overall uses of computers in instruction. The chapter concludes with recommendations for computer use whose full implementation would lead to major changes in the current instructional system. Chapter 4 presents a workshop dealing with several topics related to computers and problem solving: definition of the problem; problem solving ideas; roles of computers in problem solving; accumulated knowledge of humans; and effective problem solving procedures. Material to be used in inservice sessions for middle and secondary school mathematics teachers on integrating the computer into the curriculum--including software descriptions and lesson plans--are contained in Chapter 5. A 45-item annotated bibliography on problem solving is attached. (MES)

COMPUTERS AND PROBLEM SOLVING: AN INDEPENDENT STUDY COURSE



David Moursund
International Council for Computers in Education
1787 Agate Street
Eugene, Oregon 97403

ISBN 0-924667-40-0

Ordering Course Text

Computers and Problem Solving: An Independent Study Course

Please send me Computers and Problem Solving	$ 18.00
Add $2.50 per book for invoiced orders	$ ______
For Airmail outside of the U.S., add $10 shipping	$ ______
Total	$ ______

Payment:

☐Check enclosed

☐Visa ☐Mastercard

Name on card ____________________

Card number ____________________

Expiration date ____________________

Send this form to:

Independent Study, ICCE
University of Oregon, 1787 Agate St.
Eugene, OR 97403

Telephone orders also welcome:
503/686-4414.

COMPUTERS AND PROBLEM SOLVING: AN INDEPENDENT STUDY COURSE

David Moursund
International Council for Computers in Education
1787 Agate Street
Eugene, Oregon 97403

Contents

Chapter 1 Overview of Computers in Education

Chapter 2 Higher-Order Skills

Chapter 3 The Future of Computers in Education

Chapter 4 Computers and Problem Solving: A Workshop for Educators

4.1 Preface
What is a Problem?

4.2 Key Ideas in Problem Solving

4.3 Roles of Computers

4.4 Accumulated Knowledge of Humans

4.5 Effective Procedures

4.6 Conclusions and Recommendations

4.7 Appendix on Active Listening

References

Chapter 5 Material from National Science Foundation Project on Effective Inservice

5.1 Narrative Overview

5.2 Script

5.3 Timeline

5.4 Handouts

5.5 Materials

5.6 Readings

Appendix Problem Solving: An Annotated Bibliography by Billy C. Yates

Chapter 1

Overview of Computers in Education

Note: This chapter is adapted from a chapter in the *Computer-Integrated Instruction Mathematics Notebook*, developed under National Science Foundation grant number TEI-8550588 (David Moursund, project director). The adaptation draws heavily from a similar chapter in *Long-Range Planning for Computers in Schools*, written by Moursund and Ricketts, which was adapted from the same original source.

Brief Summary

A few schools have used computers for nearly 30 years. Thus, the discipline of computers-in-education is quite well established. The overall field can be divided into administrative, instructional and research uses. The main focus of this chapter is instructional *uses of computers in schools.*

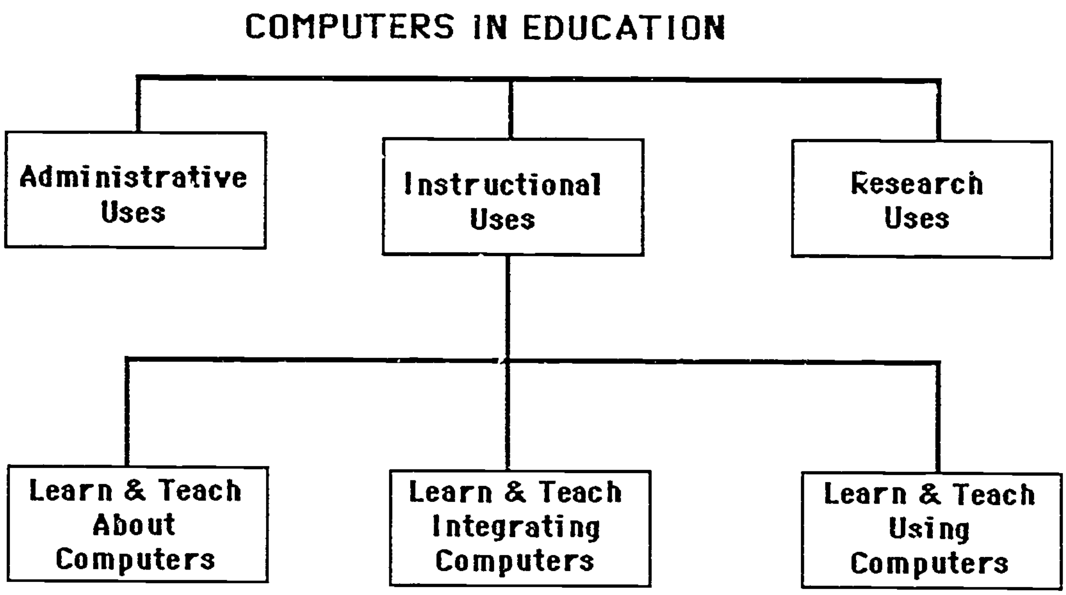
The instructional uses of computers can be divided into three parts: learning about computers (computer science, computer programming, and the general theory of computers); computer-as-tool (Computer-Integrated Instruction, or general use of the computer as an aid to problem solving throughout the curriculum); and learning using computers (Computer-Assisted Learning or Computer-Assisted Instruction).

Introduction

Computers are important and widely used in our society because they are cost-effective aids to problem solving in business, government, industry, education, etc. This chapter provides an overview of computers in education, with primary emphasis on instructional uses. The intent is to lay a foundation for use of computers as an aid to problem solving throughout the curriculum.

Computers in Education

The diagram below shows a structure of the overall field of computers in education.



As indicated in the diagram, the overall field of computers in education can be divided into three main parts. While we will discuss each part, the focus will be on **Instructional Uses** of computers. As the diagram illustrates, instructional uses of computers can be divided into three parts; later in this chapter we will discuss each part in some detail.

Administrative Uses

Running a school system is similar in many ways to running a business. That is, a school administrator faces many of the same problems as does a business administrator. A school system has income and expenses. It has facilities and inventories. It has employees who must be paid and employee records that must be maintained. And, of course, a school system has students who must be taught. Detailed records must be kept on student performance, progress and attendance. In some ways, students and their parents are similar to customers in a business.

Computers can be cost-effective aids to accomplishing all of the administrative-oriented tasks listed above. It is not surprising that computers are extensively used for administrative purposes in most Canadian and U.S. school districts. In some school districts this use goes back more than 25 years. Overall, the administrative use of computers in schools is growing steadily.

At the current time there are two major approaches to administrative use of computers in schools. One approach is centralization. A large, centrally located computer system is used to serve a number of schools as well as the district office. Some input and output operations may occur at the school terminals, but most operations, especially those involving large amounts of input and output, occur at the central facility.

An alternate, decentralized approach has gained considerable support in recent years. Administratively oriented microcomputer systems are located at individual schools. While in the past these were self-contained microcomputers, now there is a growing tendency to hook them together in a computer network. It has become clear that microcomputers can make a substantial contribution to the efficiency of a school office.

This is a key idea. Personnel in a school office are faced with problems that must be solved. Some of the problems require careful interaction with people (faculty, students) and their solutions require good interpersonal skills. Other problems lend themselves to use of computers as an aid to the storage, processing and retrieval of information. There are many relatively routine "paper shuffling" tasks where appropriate use of computers can greatly increase the productivity of office staff. There are other, more complex tasks, such as bookkeeping and order processing, where computers can be a significant aid.

It seems evident that there will be a continuing need for a central, powerful computer system in most school districts. It is important to have one small group of people be responsible for payroll, the overall budget, long-term student records, etc. Also, it seems evident that on-site microcomputers will become increasingly popular. What is not so clear is how and to what extent the central facility and the on-site microcomputers should be networked together. Nor is it always evident which computer applications are best accomplished at the school site and which are best accomplished at the central facility.

The problem of the design and implementation of a school district's administrative computer system is a task for computer professionals who can take into account hardware and software advances as well as changes in the district. It is the type of problem whose solution requires broad knowledge and experience.

Research Uses

Educational research is often directed toward solving a specific educational problem or gathering information that can help solve a particular problem. Educational research has benefited immensely from computers. Many educational research projects involve collecting a large quantity of data and subjecting it to careful statistical analysis. For example, one may have a control group and treatment groups. Students in the groups are extensively tested during various phases of the research experiment, resulting in a substantial collection of data. Large libraries of statistical programs have been available for more than 25 years. Now such program libraries are even available on microcomputers. A researcher who is knowledgeable in the use of statistical packages can easily carry out a number of statistical analyses on data that have been collected.

The use of statistical packages illustrates an important idea about roles of computers in problem solving. The type of educational research we are describing requires careful thinking when defining the problem, setting up the experiment, deciding which statistical analyses to perform, and interpreting the results of the statistical analyses. A computer can be used to perform the actual statistical analyses, and the researcher does not need to know the details of how the computer performs these analyses. Some problem solving tasks are better accomplished by machine and others are better accomplished by the human mind.

Computers are making it easier to conduct longitudinal research studies. Detailed records on students, teachers, schools, and school districts can be kept over a period of years. These records can then be analyzed for patterns or trends that might not be evident from a manual search.

Computer-Assisted Learning (which we discuss later in this chapter) lends itself to exciting research. As students interact with computers while studying a particular topic, the computers can collect and maintain detailed records. Analyzing these records helps determine which aspects of the instructional program seem most effective, and which need modification. Such formative evaluation is leading to significant improvements in the quality of Computer-Assisted Learning materials and will eventually have a profound impact on instruction.

It is important to understand that administrative, research, and instructional uses of computers are relatively distinct fields of study. A person might be an expert in administrative uses of computers, yet have little knowledge of the statistical packages and statistical techniques of a researcher. Similarly, a person might be an expert in instructional uses of computers but have little knowledge of the hardware and software needed in an administratively oriented computer system. In fact, the number of people who are simultaneously experts in any two of these fields is quite small.

The message should be clear. To solve problems in a particular discipline one needs to know both the discipline and the aids to problem solving within the discipline. A person can be quite knowledgeable in the use of computers and still know next to nothing about solving school administrative, research, or instructional problems .

Instructional Uses: Learn and Teach About Computers

The instructional use of computers can be divided into three main parts. Learn and Teach About Computers focuses on the discipline of computer science. (Here we use a very broad definition of computer science, to include computer engineering, data processing, information science, and so on.) This is a well-established discipline; hundreds of journals and magazines publish a rapidly growing body of computer-related research. The Association for Computing Machinery, a very large professional society for computer scientists, was established in 1947. Many college and university computer science departments are now more than 20 years old.

Each academic discipline can be defined by the types of problems it addressees. Computer science focuses on the computer as an aid to problem solving and a rapidly growing body of ways to use a computer as an aid to problem solving. A key idea is to develop a procedure (which can be executed by a computer) to solve or help solve a particular category of problem. Thus, computer science places a major emphasis on procedural (or algorithmic) thinking.

Research into student performance in introductory computer science courses strongly supports the contention that students who have good mathematical aptitudes and backgrounds do well in computer programming and computer science courses. In recent years college-level computer science courses have tended to become more rigorous and math-like. They tend to emphasize the types of problem solving skills that are emphasized in mathematics courses.

A few high schools began to experiment with teaching computer programming in the late 1950s. This early use of computers in schools provided solid evidence that high school students could learn to program in assembly language or FORTRAN. However, computers were quite expensive and not particularly accessible to high school students.

The development of timeshared computer systems and the BASIC language in the early 1960s opened up the possibility of large numbers of students learning to write computer programs. As timeshared computers decreased in price, more and more schools began to offer a course in BASIC programming.

By the early 1970s, it was becoming clear that computers were beginning to transform our society. The Industrial Age had ended, and the Information Age (sometimes called the Computer Age) had begun. Many educators argued that all students should become "computer literate," and that this is best accomplished through specific computer-oriented coursework. Often the courses were introductory programming instruction in the BASIC programming language. The trend toward students taking computer programming increased rapidly as microcomputers became available to schools in the late 1970s.

Now a counter trend is beginning to emerge as people realize that it is not necessary to learn to write computer programs in order to make effective use of a computer in solving a wide range of problems. Many introductory courses have reduced their emphasis on computer programming and increased their emphasis on using applications software (computer-as-tool). Computer literacy courses have been developed that contain little or no computer programming.

The rapid growth of applications-oriented computer literacy courses have caused a number of educational leaders to ask, "Why limit such instruction to a specific course? Might it be better for students if computer applications were taught throughout the curriculum?" The idea is that students should make use of computer-as-tool in all courses where this is appropriate. That is exactly what Computer-Integrated Instruction (CII) is about. We will discuss CII more later in this chapter.

The teaching of computer programming and computer science courses at the precollege level is slowly beginning to mature. A Pascal-based Advanced Placement course has been developed and is now widely taught. This has tended to structure the high school computer science curriculum. However, it is evident that this type of course appeals to only a small percentage of high school students. Enrollment in courses that use BASIC, Logo, or other non-Pascal-like languages remains high. However, total enrollment in secondary school computer science and programming courses seems to have peaked and and appears to be declining.

Logo has developed a wide following, particularly in elementary schools. Some Logo-oriented teachers view the learning of Logo as an end in itself, but most recognize the potentials of Logo as a vehicle for illustrating and teaching various problem solving strategies and some content. Also, the Turtle geometry part of Logo can help students learn a number of important geometric ideas.

A number of research studies have tried to show that students who study Logo naturally and automatically develop improved problem solving skills in other areas. Most of these studies do not support such a contention. It appears that there is little "natural" transfer of problem solving skill from a Logo environment to other environments. A good teacher can greatly increase this transfer of learning of problem solving skills.

Instructional Uses: Learn and Teach Integrating Computers

The second category of instructional use of computers is Computer-Integrated Instruction (CII). It focuses on the computer as a productivity tool--as an aid to problem solving or getting work done. One aspect of CII is use of general-purpose application packages such as database, graphics, spreadsheet, word processor, and telecommunications. Each of these application packages is widely used in business, industry and government. In education, each can be used at a variety of grade levels and in a variety of courses.

A second aspect is developing applications software specific to a particular discipline. For example, a substantial body of software is designed to help a person compose music. Such software makes possible the teaching of musical composition to grade school students--which would be a significant change in the music curriculum.

It has long been recognized that precollege students could learn to make use of computers as an aid to problem solving. The initial approach, now dating back more than 25 years, was to have students learn to write computer programs to solve specific categories of problems. For example, it was suggested that writing a computer program to solve quadratic equations in a second year high school algebra course indicated real understanding of that mathematical topic.

Initially such an approach to CII made little progress because both the programming languages and the computer hardware were not suited to most precollege students. But the advent of timeshared computing and BASIC helped change that situation. And then microcomputers, with built-in BASIC, made it feasible for millions of students to learn to write simple programs to solve specific categories of problems.

However, it takes considerable time, as well as specific talents, to become a competent computer programmer. Educators and students soon recognized that the time was being taken away from the study of conventional subject matter. The movement toward integrating computer programming into various high school courses has long since peaked and has been replaced by a trend toward using applications packages. This new trend has accelerated as better applications packages have become available for the types of microcomputers used in schools. An increasing percentage of this software is specifically designed for use in education.

Word processing can be used to illustrate both the general idea of CII and some inherent difficulties. Word processing is a "generic" computer application tool, in the sense that it is applicable across the entire curriculum, at all grade levels. It is quite clear that a word processor is a cost-effective productivity tool for secretaries and for many people who do a lot of writing. Moreover, word processors make it easier to do process writing (prewrite, compose, conference, revise, publish). For these reasons, some schools have decided to have all of their students learn to do process writing in a word processing environment.

But it takes quite a bit of instruction to learn to make effective use of a word processor. To learn proper keyboarding techniques and to keyboard faster than one can handwrite takes a typical fourth grade student about a half-hour a day for eight weeks or more. To learn to compose at a keyboard and make effective use of a word processor takes additional instruction and practice.

There are several additional difficulties. First, teachers have to learn to provide the initial instruction and to work with students who do process writing in a word processing environment. Even if the initial instruction is provided by a specialist rather than the regular classroom teacher, the regular teacher must work with students after the initial instruction. All of the students' subsequent teachers face the same problem. This suggests that large numbers of teachers will need to learn to work with the idea of process writing in a word processing environment.

Second, there is the matter of access to appropriate computer systems. Once a student becomes adept at this mode of writing, the student will want to continue its regular use. This can easily require providing each student with a half-hour of computer time per day, a ratio of one computer for every 10 to12 students, for this use alone. It also raises the issue of needing to provide computer access to students at home after school and on weekends. Thus, while computers can help address the writing "problem," in doing so they create other problems such as equity. The nature of real world problem solving is that the steps one takes to solve a particular problem may create other problems.

Third, there is the problem of testing--especially standardized testing. Suppose a student has had some years of using a word processor. The student has learned to approach writing projects using this productivity tool. There is a good chance the student can write better and faster using a word processor than using pencil and paper. An appropriate assessment of this student's writing skills requires giving the student access to a computer during the test.

Fourth, once one has a word processor, it is quite helpful to also have a spelling checker and a grammar/style checker. Such aids to writing may have a significant impact on the nature of the writing curriculum. They may require changes in textbooks, lesson plans, and the way class time is structured. And once again the issue of testing arises. Should a student be allowed to use spelling and grammar checkers when doing writing for an essay test?

These four types of difficulty--training, access, testing, and curricular change--occur for all CII applications. The problem of teacher training is gradually being addressed by colleges of education, and most school districts have expended considerable resources to help their teachers become computer literate. The problem of access to appropriate hardware and software will be with us for some time. It can be overcome through appropriate allocations of money. The testing problem is being addressed by a number of agencies involved in widespread assessment. For example, some states and provinces now allow use of calculators on certain tests. However, it seems clear that this will be a long-term problem. Textbook companies are slowly beginning to address the issue of integrating computer-as-tool into their product lines. School districts and individual teachers interested in making more rapid progress are developing their own curriculum materials.

Instructional Uses: Learn and Teach Using Computers

The third major category of instructional uses of computers is use of computers to help deliver instruction. This use has been called computer-assisted instruction, computer-based instruction, computer-assisted learning, etc. We will call it Computer-Assisted Learning (CAL), which puts the emphasis on the hoped-for results.

CAL is sometimes divided into categories such as drill and practice, tutorials, and simulations or microworlds. Most CAL systems include a record keeping system, and some include an extensive diagnostic testing and management system. Thus, computer-managed instruction (CMI) is sometimes considered to be a part of CAL.

Initially most CAL material was designed to supplement conventional classroom instruction. For example, elementary school students might use mathematics drill and practice materials every day for 10 minutes. (Much of the initial research on CAL conducted during the late 1960s and early 1970s was based on a paradigm of 10-minute drill sessions.) But as computer hardware costs have declined and more CAL materials have been developed, there is some trend toward the implementation of substantial units of study and/or entire courses. Declining hardware costs are making such CAL economically feasible. Indeed, in some situations such courses are now less expensive to offer than teacher-taught courses.

CAL has been heavily researched over the past 30 years. The evidence strongly supports the educational value of using CAL in a wide variety of settings. Much of the success of CAL may be explained by three factors:

1. Students using CAL (on average) spend more time on task. Since learning correlates well with time on task, students (on average) learn faster using CAL.

2. Students work with materials appropriate to their ages and specific needs. This individualization of instruction increases efficiency.

3. CAL materials can incorporate good practices of instructional and learning theory. Formative evaluation can provide a basis for improving CAL materials under development. Through this approach, the quality of commercially available CAL materials is gradually being improved.

Computer simulations in education are of growing importance. They can help create educationally sound learning environments that are fun, exciting, and conducive to learning. In essence, a simulation creates a problem solving environment in which students can practice a variety of problem solving techniques. Some simulations are used by individuals or small groups. Others are used by the whole class, with the teacher facilitating the class' interactions. Needless to say, such uses of computers require changes in the way most classes are usually run, and they require quite a bit of staff development. Also, they require computer access in the regular classroom.

CAL holds great potential. It facilitates individualization of instruction and increased educational opportunities. It facilitates learning at home and on the job as well as while at school. It is clear that CAL will be of steadily growing importance in our schools. Long-range planning must address appropriate roles of CAL and assume that CAL software will continue to improve.

Computer-Integrated Instruction and Problem Solving

Many work environments now include a computer or computer terminal for every employee. This will become more and more common, since computers are such useful aids to solving problems and increasing human productivity. You can assume that increasing numbers of today's students will use computers when they go to work. (A 1986 estimate indicated that 14 percent of the labor force in the United States was using general-purpose computers regularly on the job.)

Research on transfer of learning strongly supports the position that instruction and training should closely parallel the final desired behaviors. If we need workers adept at using computers to aid in solving problems, we should integrate computer use as students develop their basic problem solving skills and strategies. For these and other reasons, CII will grow rapidly for many years to come.

As CII increases, both teachers and students will begin to question the content of many of their courses. **If a computer can solve or help solve a particular type of problem, what should students learn about the problem?** Is it necessary and appropriate to learn to solve each type of problem using only Industrial Age aids such as book, pencil and paper? Or should schools focus more on underlying concepts and help students gain an overall understanding of problems that computers can solve?

In some cases an answer will be forced on schools. Libraries are being computerized. Card catalogues are being replaced by computerized information retrieval systems. Important publications and data are available only in computer databases. Since learning to access information is an essential component of education, students will have to learn to use databases and computerized information retrieval systems.

In other cases schools will have many options. For example, consider the impact that handheld calculators have had on the upper elementary school and middle school mathematics curriculum. While the potential for calculator-integrated instruction is large, the actual impact on the curriculum has been minimal. This is true in spite of 10 years of strong support from the National Council of Teachers of Mathematics for integration of calculators into the curriculum. In April 1986 the NCTM issued still another strong statement recommending calculator use at all grade levels.

Much of the short-term potential for CII depends on how well our educational system addresses the issue of inservice education. All current teachers can learn to make effective use of CII. Given appropriate inservice educational opportunities and access, many will do so. Every school and school district should have a staff development plan that provides appropriate opportunities for all teachers to become competent users of computer-as-tool.

References

Bork, Alfred. *Personal Computers for Education*, Harper & Row, 1985.

Chandler, Daniel. *Young Learners and the Microcomputer*, Open University Press, 1984.

Gebhardt-Seele, Peter A. *The Computer and the Child: A Montessori Approach*, Computer Science Press, 1985.

Merrill, Paul F.; Tolman, Marvin N.; Christensen, Larry; Hammons, Kathy; Vincent, Bret R.; and Reynolds, Peter L. *Computers in Education*, Prentice-Hall, 1986.

Moursund, David. *Introduction to Computers in Education for Elementary and Middle School Teachers*, ICCE, 1983.

Moursund, David and Ricketts, Dick. *Long-Range Planning for Computers in Schools*, Information Age Education, 1987.

Moursund, David et al. *Computer-Integrated Instruction Inservice Mathematics Notebook.* (To be published by ICCE in early 1988.)

Nathan, Joe. *Micro-Myths: Exploring the Limits of Learning with Computers*, Winston Press, 1985.

Taffee, Stephen J., Ed. *Computers in Education, Second Edition*, Dushkin Publishing, 1986.

Taylor, Robert, Ed. *The Computer in the School: Tutor, Tool, Tutee*, Teachers College Press, 1980.

Chapter 2

Higher-Order Skills

Note: This chapter is a slightly modified version of Chapter 4 of *High Tech/High Touch: A Computer Education Leadership Development Workshop* written by David Moursund. The book consists of 15 chapters and is written as a sequence of workshop sessions. Each workshop session is about two to four hours in length. The workshop involves a considerable amount of small group interaction and focuses on the balance between people-oriented skills and technically oriented skills. Workshop participants have already received instruction and practice in Active Listening. By this stage of the workshop they are quite comfortable in interacting both in small group and large group discussions. The written materials attempt to capture the spirit of being in such a workshop setting.

Goals

1. To increase understanding of the "back to basics" movement and of the issues involved in lower-order versus higher-order skills.

2. To understand the potential impact of computers on the current balance between lower-order and higher-order skills in our educational system.

3. To examine ways to make use of computers in increasing higher-order skills.

Bloom's Taxonomy

Within each academic discipline there is a continuum of knowledge and skills. This continuum can be viewed or divided in a number of ways. For example, Bloom divides the continuum into (1) knowledge, (2) comprehension, (3) application, (4) analysis, (5) synthesis, and (6) evaluation (Bloom, et al., 1956). Others merely talk bout lower-order and higher-order skills.

The "basics," or lower-order end of the knowledge continuum, for a particular discipline consist of the facts and skills that serve to define the discipline and allow one to recognize and solve some simple, low-level problems of the discipline. In discussing these lower-order facts and skills one might emphasize terms such as recall, training, speed, and accuracy. In mathematics, for example, the one-digit addition facts are considered to fall on the lower end of the Bloom's taxonomy scale.

The high-order end of the continuum of knowledge and skills within a discipline is discussed using such terms as analysis, synthesis, interpretation, evaluation, inquiry, reasoning, and problem solving. These are often called thinking skills. In this session we will use the terms thinking skills and higher-order skills interchangeably.

WHOLE GROUP EXERCISE: Compile a list of terms used in talking about lower-order skills; compile a second list used in talking about higher-order skills. This can be done by putting the headings Lower-Order Skills and Higher-Order Skills on an overhead projector. Then have workshop participants make contributions to first one list, then the other. The purpose of the exercise is to get participants involved and actively thinking about lower-order versus higher-order skills. (Note to people enrolled in the Independent Study course: An activity such as this can be used with students at a variety of grade levels and in a variety of disciplines. The idea is to get students to begin to think about thinking and to think about learning. This metacognition is an important aspect of education for improving higher-order skills.)

WHOLE GROUP EXERCISE: Have the participants in the workshop think about how much time they have their students work on lower-order skills and how much time on higher-order skills. The result is to be expressed as a percentage of time spent on lower-order skills, rounded to the nearest 10 percent. Then gather summary statistics from the participants by asking for data from elementary school teachers, middle school or junior high school teachers, and senior high school teachers.

Lower-Order and Higher-Order Skills

In recent years there have been a number of studies and reports recommending that schools place increased emphasis on higher-order skills. Goodlad, for example, comments on the irony of how much social studies instruction emphasizes lower-order skills when instead it could be developing reasoning skills such as "deriving concepts from related events, testing in a new setting hypotheses derived from another set of circumstances, exploring causal relationships, drawing conclusions from an array of data..." (Goodlad, 1983).

The Paideia Group analyzes disciplines in terms of (a) acquisition of organized knowledge, (b) development of intellectual skills--skills of learning, and (c) enlarged understanding of ideas and values (Adler, 1982). The third classification emphasizes higher-order skills, and the Paideia Group recommends that this classification receive increased attention.

Summaries of a number of books and reports arguing for increased emphasis on higher-order skills are give in a recent Educational Development Center book (Felt, 1985). Summaries of 11 recent articles on improving students' thinking skills are given in a publication by ERIC (1984). The general flavor of such literature is that that leading educators believe our educational system should be placing increased emphasis on problem solving. Most of this literature suggests that we have placed too much emphasis on lower-order skills during the past 15 years.

WHOLE GROUP EXERCISE: 5 minutes. Pick a discipline you know well. Make a brief list of some of the main lower-order skills in the discipline; make a brief list of some of the higher-order skills. The purpose of this exercise is to help workshop participants understand that they can readily differentiate between lower-order and higher-order skills.

DEBRIEF: Ask if anyone had problems differentiating between lower-order and higher-order skills. If so, have them share their examples.

One way to view the lower-order versus higher-order issue is that the lower-order facts and skills are the fundamental building blocks one repeatedly uses when carrying out higher-order processes. This type of analysis can be used to lend support to the "back to basics" movement. The argument is that one cannot expect to acquire higher-order skills without having the lower-order skills. "You've got to learn to crawl before you can learn to walk. You've got to learn to walk before you can learn to fly."

Both common sense and substantial research support this position (Fredericksen, 1984).

The back to basics movement received considerable attention and support during the 1970s and early 1980s. National and state assessments indicate that progress actually occurred in improving scores on tests of the basics. That is, overall declines in large scale assessments (such as the SAT scores) represent substantial declines in higher-order skills that were only partially offset by increases in lower-order skills (Felt, 1985, page 27). In essence, as all teachers know, we can "teach to the test." If national assessment is going to emphasize lower-order skills, our school systems will emphasize lower-order skills and test scores in these areas will go up.

During this same time span we have begun to obtain some usable international assessment results. National and international results from mathematics assessment have proven quite discouraging to mathematics educators in the United States (Carpenter et al., 1981; Travers and McKnight, 1985). Japan, for example, has a higher high school completion rate than the United States. Although top U.S. students do as well as top Japanese students in math achievement, the mean achievement scores on the entire population strongly favor the Japanese.

WHOLE GROUP EXERCISE: 5 minutes. Pick a discipline you you know reasonably well. Make two lists of basics or lower-order skills. The first list is to be specific to the discipline you select. The second list is to be skills that are not *specific* to the discipline, but *essential* to the discipline. Here we are looking for basic skills with high transfer value. The ability to use a dictionary to check the spelling or definition of a word might fall into this latter category. Reading words has high transfer value while reading music has much lower transfer value.

GROUP DISCUSSION: Using ideas from workshop participants, create a list of basic skills that are not specific to any particular discipline. For each item on the list, check to see if some people also included it within their specific discipline list. Finally, for each item on the list, discuss whether computers and other related technology are having or will have any effect. Look for general patterns of how computers might make a difference in basic skills. Are there differences that are suggested pedagogy or content?

While the past decade has seen increased emphasis on basics, general educational research and progress in technology have been whittling away at this extreme emphasis on lower-order skills. Educational leaders have argued that there is too much emphasis on basics (Sizer, 1984). "In a recent poll of professional educators, nine out of 10 respondents said that better instruction in thinking skills should be a priority in educational planning for the coming years," (Beyer, 1984). Moreover, there has been substantial progress in understanding problem solving and how to help students become better problem solvers. An excellent review of the literature on teaching problem solving is provided by Fredericksen (1984). A key point is that students become better at problem solving in a particular discipline if they are given specific instruction and practice in problem solving in that discipline. As with lower order skills, students learn what is taught.

Transfer of Knowledge and Skills

For simplicity, let's keep in mind a "reading, writing, arithmetic" definition of basics. To a large extent, we see:

1. There is use of these basics in many disciplines.
2. There is transfer of learning of basics.

The latter point is quite important. For example, basic arithmetic skills learned in a math class are apt to be used by students in non-math classes. Basic reading skills obviously transfer to other disciplines, even though there is need for special instruction in how to read technical materials such as math and science.

There is substantial literature on transfer, and we won't go into it here. Roughly speaking, the literature indicates that the more closely alike two situations are, the more transfer will occur between the two. (The literature talks about near transfer and far transfer. If one learns to tie one's left shoe it is a near transfer to learn to tie the right shoe. If one learns metric measurement in a math class, it may be a far transfer to using this knowledge in a science class, and still farther to using it in a setting outside of school.) If one wants to teach for transfer, give many examples from a wide range of possible applications of the knowledge and skills being studied.

Transfer of learning is one of the most important ideas in education, and it is also a poorly understood idea. It seems desirable to include in our educational system basic knowledge and skills that readily transfer. But many people have a mental image of a mind as a muscle, and that exercising the mind develops its general capabilities. For example, they tend to believe that studying geometry, algebra and higher math makes one into a better thinker and overall problem solver. Earlier generations of educators used the same arguments to support the study of Greek and Latin. There is little evidence to support this position, and it has largely been discarded by educational leaders.

But we now have a new generation of computer educators who feel that the teaching of problem solving in a computer programming environment will make students into better problem solvers. Not surprisingly, research evidence to support this contention is sparse. Indeed, even though mathematics problem solving and problem solving in a computer programming environment seem closely related, relatively few studies have found transfer from computer programming to math problem solving skills. Note, however, that high scores on mathematics aptitude tests correlate highly with achievement in computer programming and introductory computer science courses.

INDIVIDUAL AND TRIAD GROUP EXERCISE: 10-15 minutes. Each individual is to select a specific discipline and make a brief list of higher-order skills for that discipline (five minutes). Then, working in triads, discuss the lists. Look for examples of higher-order skills specific to one discipline that seem to transfer easily to another discipline. Look for higher-order skills that seem quite specific to a single discipline, with little transfer value.

DEBRIEF: Via group sharing, make a list of examples of higher-order skills that transfer. Look for patterns, or for some general theory that would help explain transference or lack thereof. Make sure that the discussion includes an examination of which of these high-transfer, higher-order skills are emphasized in our educational system. Do computer-related examples get mentioned?

Computer Simulations

The debriefing discussion for the previous exercise may move in the direction of simulations or computer simulations. Research into use of educational simulations supports the idea of giving students opportunities to practice the skills they will later be expected to perform. For example, many of the skills required to fly an airplane can be practiced in a flight simulator. Good transfer of learning occurs because the flight simulator is very similar to a real airplane. (That is, it is a near transfer from the flight simulator to the real airplane.) Similarly, much of modern laboratory science makes use of very sophisticated instrumentation--equipment that is beyond the financial reach of most high schools. But a number of excellent simulations exist that can help students gain a deep understanding of a variety of fundamental experiments. Their use is supported by science educators (Steidley, 1983).

During the past 20 years there has been substantial research on educational simulations covering a broad range of traditional school topics. The computer-based educational simulations developed under an NSF project directed by Ludwig Braun in the early 1970s (the Huntington Project) provided excellent examples of what could be done with the computer equipment available to schools at that time. Now an increasingly wide range of computer-based educational simulations are becoming available. A recent Ph.D. dissertation at the University of Oregon contains an excellent survey of the literature and an in-depth study of a simulation designed to help students learn key ideas in personal health (Woodward, 1985). The dissertation study supports the contention that a computer simulation can be used to help students gain applicable skills in the personal health area. In Woodward's study the simulation was used as a supplement to conventional instruction, covering and reinforcing the same materials. This seems to be an effective approach.

WHOLE GROUP DISCUSSION (IF TIME PERMITS): Ask workshop participants to share some of their personal experiences in making use of computer simulations with students. What evidence can the workshop participants provide on the efficacy of using the simulations? (As an example, many educators are familiar with the *Oregon Trail* simulation from MECC. What do students learn from using this simulation?)

While many computer educators strongly support use of simulations, few can cite relevant literature to support such use. Instead, they have a "gut level" or "intuitive" feeling that it is good to use simulations. In essence, the simulations have fac validity.

Three Examples

The issue of what constitutes an appropriate balance between instructional emphasis on lower-order skills and instructional emphasis on higher-order skills is crucial. In this section we provide three examples; in each case we see how technology (especially computer based technology) is entering the picture.

GROUP DISCUSSION EXERCISE: The three examples given in the next few pages are:

1. Writing.
2. Mathematics.
3. Information retrieval (databanks).

Divide workshop participants into groups of perhaps 10 people each, with each group selecting one of the three examples. Each group is to spend about a half-hour discussing the ideas of the example. Major emphasis should be placed on higher-order skills (problem solving) in the discussion. Add to and/or expand the ideas. Discuss the likelihood that the changes being suggested will occur--and if so, when. What can computer education leaders do to hasten the proposed changes? Should they do this?

EXAMPLE 1 (WRITING)

The overall process of writing involves some lower-order skills such as spelling, grammar, and handwriting. But in total, writing is a higher-order process, requiring use of many higher-order skills. Both prewriting and writing involve careful and reasoned analysis, interpretation, and organization of ideas. Good writing involves repeated revision based on feedback provided by oneself and others. If writing is evaluated holistically with emphasis on higher-order processes, then a program of study emphasizing spelling, grammar, and handwriting does little to improve a student's writing. Note also that the issue of time spent on teaching/learning cursive writing is important. Time spent on cursive writing could be spent learning keyboarding and in practicing writing.

Technology has produced the typewriter and the word processor. Technology has produced the computerized spelling checker and computerized grammar checkers that can run on a microcomputer. Research has produced the idea of process writing--for example, as taught through the Bay Area Writing Project. Process writing includes prewriting, composing, revision (using feedback generated by self and others) and presentation of the final product.

Recently, the technology and research in writing have begun to combine. Students are learning to do process writing in a word processing environment. Two doctoral students at the University of Oregon recently completed their dissertations in this area (Boone, 1985; Wetzel, 1985). They have developed a successful model for helping teachers learn to teach process writing in a word processing environment. It seems likely that eventually all students will enter secondary school having grown up in a process writing environment; they will be reasonably skilled at using a word processor. Meanwhile, this is a problem situation that must be attacked at both the elementary and secondary school levels. For many years to come the majority of secondary school students will lack these word processing and process writing skills.

Please read the last paragraph of the next section, as it is also applicable to this section.

EXAMPLE 2 (MATHEMATICS)

In mathematics, lower-order facts and skills include vocabulary, notation and computation. The elementary and middle school mathematics curriculum places substantial emphasis on students developing both accuracy and speed in paper-and-pencil computational skills. The core of the grade school mathematics curriculum is built around such computation. Generally, performance in computational tasks is a key determiner as to the mathematics track students follow in junior high and high school. (Note that many quite good mathematicians have relatively poor lower-order skills in mathematics. It may be that placing emphasis on lower-order skills actually screens out a number of people who could become quite good mathematicians.) Thus, tremendous amounts of time are spent on computation. Grayson Wheatley, a well known mathematics educator, estimates that the typical U.S. student completing the ninth grade has spent 200 hours studying and practicing paper-and-pencil long division of multidigit numbers (Wheatley, 1980).

Teaching students to add, subtract, multiply and div; fractions also receives considerable attention in the mathematics curriculum. Wheatley's estimate on time spent on long division may also be a good estimate of time spent learning to calculate with fractions. (Note that if one uses a metric system of measure, the need to calculate with nondecimal fractions is greatly reduced. A complete switch to metric would allow a substantial decrease in emphasis on calculation with fractions.)

Now, of course, we have the calculator and the computer. For many years, the National Council of Teacher of Mathematics (NCTM) has recommended increased use of calculators and computers (NCTM, 1980). Recently NCTM has come out with the even stronger recommendation that the entire math K-12 math curriculum should incorporate use of and adequately reflect the capabilities of calculators and computers (NCTM, 1985). The recommendation is that much less time be spent on lower-order, paper-and-pencil computational skills, and that more time be spent on higher-order skills. Problem solving is to be emphasized throughout the mathematics curriculum.

These same NCTM reports point to the fact that students do not do well in transferring their mathematical knowledge to other subject areas. For example, many students do not seem to realize that mathematical techniques are important in the social studies. Social studies courses tend to make relatively little use of the graphing, equation-solving, or mathematical modeling techniques students learn in mathematics classes. Sizer addresses the fact that our secondary school system is quite discipline oriented, whereas real-world problems tend to be interdisciplinary (Sizer, 1984, page 133).

It seems likely that eventually all secondary school mathematics classes will meet in a computer lab, with perhaps one computer per student. Moreover, eventually students will enter secondary school having had many years of experience in using calculators and computers as aids to learning mathematics and as aids to doing mathematics. If current trends continue, many of these students will have had substantial experience with Logo while in elementary or middle school.

Mathematics is a particularly interesting area for discussing the potential impact of computer-based technology on the lower-order versus higher-order skills area. From the point of view of a fifth grade student or teacher, the long division of multi-digit numbers is a higher-order task. But it takes only a few minutes of instruction in using a calculator for a typical fifth grade student to acquire long division skills exceeding the average paper-and-pencil skills of students who are several years older. that is, this higher-order task becomes a lower-order task through the use of a calculator.

The calculator and long division example is just the tip of an iceberg. In calculus, for example, students learn to differentiate and integrate a wide range of functions. But computer programs (for example, *Mu-Math*, which runs on a microcomputer) can accomplish such tasks. Indeed, there are now hand held calculators available that can accomplish many of these tasks. These calculators can also graph functions and automatically solve equations. From the viewpoint of a student just beginning differentiation and integration, these are certainly higher-order skills. But from the viewpoint of an accomplished mathematician, they are lower-order skills. The fact that they can be computerized both lends credence to their classification as lower-order skills and allows of a shift of emphasis in this area of mathematics education. Some of the time spent developing speed and accuracy in by-hand differentiation and integration could be spent developing higher-order problem solving skills in calculus.

In both the writing and the math examples, the overall goal is to have students develop improved higher-order skills. In both cases there is research that suggests increased emphasis on lower-order facts and skills does little to improve the higher-order skills. In both cases technology is serving as a change agent or motivation for considering change. In both cases computers can carry out or help carry out some of the key lower-order tasks. In both cases the computer technology is not central to the disciplines but provides a strong excuse for reconsidering what is currently done in school and enables significant change. The disciplines of writing and mathematics remain as core disciplines in the curriculum.

EXAMPLE 3 (INFORMATION STORAGE AND RETRIEVAL)

An excellent third example is provided by the general area of information storage and retrieval. Here we work with the basic idea of learning facts versus learning where and how to retrieve the facts. The memorization of some facts is essential to everyday functioning and to developing higher-order skills in information retrieval. But easy access to large quantities of stored information (such as in a large library) tends to increase the value in having good higher-order skills in retrieving information and using retrieved information. That is, the balance between memorization of facts and acquiring thinking skills that make use of facts is shifted by easy access to information.

This balance underwent a major shift when reading and writing were first invented. It underwent another major shift when movable type for the printing press was invented. Reading and writing provide a supplement to memorization and oral communication.

Now we have Computer-Based Information Systems (CBIS). This discipline has made very rapid progress over the past 20 years. Gerald Salton has been a leader in the field during this time. It is interesting to compare his 1968 and 1983 books, which contain broad coverages of the field (Salton, 1968; Salton and McGill, 1983). There has been substantial progress both in the underlying theory as well as in hardware and software. CBIS is now a major field in computer science and an important part of modern librarianship studies.

The hardware progress in CBIS is, of course, the driving factor. For example, in 1965 the University of Oregon was quite proud of its newly installed IBM 360/50 computer that cost about a million dollars. It had two 5-megabyte magnetic disk drives, along with the more common magnetic tape drives. Now many purchasers of microcomputers are including 10-megabyte or 20-megabyte magnetic disk drives. The manufacturer's cost of a 10-megabyte disk drive is about $200! One can add a 10-megabyte disk drive to many microcomputer systems for about $500.

IBM is now marketing a magnetic disk storage system that will store five billion characters of information (a gain by a factor of 1,000 over the past 20 years.) Five billion characters is roughly equivalent to the total contents of a typical elementary school library. But the whole storage system is smaller than a typical desk. IBM expects to sell many thousands of these magnetic disk storage systems.

Progress in laser videodisc technology for databases is even more exciting from an educational viewpoint. A laser videodisc is not erasable. [Note: Write Once Read Many (WORM) laser videodisc systems have been on the market for a couple of years. In 1987 they became available for microcomputer systems. Videodisc systems with erase/write capabilities are just beginning to come to market. Currently they are reasonably expensive and not quite as reliable as one might like.] But the videodiscs are quite cheap to produce in quantity. That is, once a laser videodisc has been produced, copies can be made by a stamping process not unlike that used to mass produce phonograph records. A single videodisc, smaller than five inches in diameter, can store 500 million characters--the equivalent of 500 long novels. The cost of producing multiple copies is perhaps a dollar or two apiece.

Laser videodisc technology has already entered a mass production stage for the home hi-fi music system market. A medium quality laser videodisc player for digitized music now retails for under $300. Laser discs for storage of computer data use almost the same technology. While some additional expense is entailed in connecting such a laser videodisc system to a computer in order to access databases stored on videodiscs, it is evident that this technology will provide a major breakthrough in the cost of storing and retrieving large amounts of information. A half dozen manufacturers expect to begin marketing microcomputer laser videodisc systems in late 1985 or early 1986, with prices expected to be under $1,000 excluding the computer. Indeed, the Atari system is supposed to be priced in the $500 range. [Note: As of March 1986, this prediction is not proving to be very accurate. A laser disc system is now available for the IBM PC and for the Apple IIe. The former costs in the range of $1,800. The latter costs even more, since it requires adding a substantial amount of memory to the Apple. There is still talk that a cheaper system will "soon" be available for the Atari. However, the rumored price has gone up several hundred dollars.] [Note: As of May 1987 the previous note still holds. The commercial market for laser videodisc systems has been slow to develop. A number of large and useful databases are now available for these laser systems. But their overall impact on education has been nil.]

Along with rapid progress in the storage of large databases has come rapid progress in communication to access databases. A combination of space satellites and fiber optics is making it possible to transmit more and more data at lower and lower costs. A March 1986 ad from AT&T indicated that, in an experiment, fiber optics were used to transmit 20 billion bits of information per second. A pair of these optic fiber wires could be used to carry on 300,000 simultaneous telephone conversations. [Note: Similar AT&T ads in 1987 suggest that a single fiber optic will carry 10,000,000 simultaneous phone conversations by the year 2000. An entire, full length, color movie could be transmitted over such a fiber optic in under one second. Rapid progress is occurring in this area of research and development.] Over the next couple of decades the copper telephone wires coming into our houses will gradually be replaced by fiber optic (glass) wires. This will provide greatly increased rates of information flow. For example, interactive television (very high quality picture phones) would be possible.

Artificial Intelligence

As we move out of the industrial revolution age and into the information age, it is evident that things will never be the same again (Toffler, 1980). Automated factories and computerized robots will take over more and more of the manufacturing processes. Computers will become everyday tools of more and more people who work with information.

As pointed out in the third example of the previous section, it is obvious that computer technology can be used to store databases and that this is making an important contribution to the information age. What is less obvious, but perhaps more important, is that a database may consist of computer programs. That is, there is an ever increasing collection of computer programs designed to solve quite complex problems. One can use a computer both to store and to execute the programs.

In the past, the programs that could be stored, retrieved, and used were designed to accomplish relatively specific tasks. For example, there are computer programs to draw a graph, solve various types of equations, compute statistical analyses, or project a company's sales based upon previous sales data. But over the past few years, computer science researchers have made substantial progress in artificial intelligence. In particular, they have developed the idea of "knowledge-based expert systems." These are computer programs that contain and use the knowledge of a human expert or a collection of human experts. The potential of such artificially intelligent computer systems is so immense that it has received international attention. The "fifth generation" project in Japan has led to major responses in both the United states and in Europe (Feigenbaum and McCorduck,1983).

In the United States there is a number of companies that produce knowledge-based expert systems. A few of these systems are now routinely used in some hospitals for medical diagnostic work. Others are used to prospect for minerals, aid in drilling oil wells, and help design advertising campaigns. A recent article in *The Wall Street Journal* indicated that about 50 systems are currently in use and another thousand are at various stages of development (Davis, 1985). [Note: A book published in early 1987 contains a discussion of 1,000 expert systems currently in use. Other literature suggests that the number of systems in use is tripling each year.]

Progress in artificial intelligence means that more and more higher-order problem solving tasks will be accomplished by merely retrieving an appropriate computer program and feeding data to that program. That is, computers make many higher-order tasks into lower-order tasks. Examination of the potential impact on mathematics education alone, for example, has hardly begun. An excellent summary of some of the key ideas is provided in the recent NSF-supported work of James Fey at the University of Maryland (Fey, 1984). Our educational system has not yet begun to cope with the idea of an "inverted" mathematics curriculum, with students learning to use sophisticated software to accomplish higher-order tasks, and not learning the underlying basic skills or theory.

At the same time we are coming to understand limitations of computers and we know many tasks that people do much better than computers. Writing, speaking, and interacting with people all provide excellent examples. In a one-on-one conversation, for example, non verbal communication skills often are dominant. These are uniquely human skills.

WHOLE GROUP EXERCISE: Make a list of higher-order skills that seem likely to remain in the province of humans (that is, skills at which humans are far superior to computers) for many years to come.

DEBRIEF: This is, of course, a high tech/high touch exercise. We are looking for high-touch examples that take unique advantage of human problem solving skills. While computers have been programmed to produce certain types of art, music, and poetry, in no sense have computers begun to approach human skills in these areas.

Another area of human dominance is interdisciplinary problem solving. Listen to a group of professors discussing research and development to improve education. The discussion may range over many academic disciplines, grade levels, and types of schools. It may cover budgeting, teacher training, parenting, the influence of television, and the role of computers. Such a discussion vividly illustrates higher-order skills. Such skills can be supplemented, but clearly cannot be supplanted, by access to computers.

Summing Up

Our current educational system was, in essence, developed before the Information Age began. Even now, relatively few schools have as many as one computer work station per 12 students. (That is, very few students have access to a half-hour of computer time per day.) [Note: Estimates suggest that during fall term of 1987 there was about one microcomputer per 23 students in the U.S. precollege educational system.] But continued rapid progress in the design and manufacture of computer hardware suggests that computers will eventually be ubiquitous (Wilson and Ticer, 1985). That is, today's first grade students will live their entire adult lives--work, play, and study--with easy access to computers with perhaps a hundred times the capabilities of the microcomputers that are now popular in our schools.

If we assume that eventually computers will be an everyday tool, we must ask where students will learn to make effective use of the tool. For example, the use of computers as a graphics aid might be taught in a mathematics course. But graphics is useful in many disciplines; students should graph data in social studies courses as well as in science courses. Computer graphics is important in graphic arts work and in technical drawing.

The point is that computers are an interdisciplinary tool. While initial instruction in use of a specific piece of software might be relegated to a particular course, students need instruction and practice in using the software in many courses. It cannot be assumed that transfer of learning will automatically occur. This creates an immense teacher inservice problem. Eventually all teachers will need to know how to work with students who have grown up with computers and who use computers as an everyday tool. All teachers will need to know both those aspects of computers that are most relevant to their own academic disciplines and how to help students use computers as general-purpose tools in their discipline.

Research is needed on effective inservice for accomplishing this teacher training task. A start is provided by the research of Ferres (1983). His research provides a model for effective inservice of elementary school teachers who want to use Logo in their classrooms. The

Computer-Integrated Instruction Session in this book discusses a National Science Foundation inservice education research project currently being conducted by Dave Moursund. This three-year project is developing models and materials for effective inservice for integration of computers as everyday tools into the curriculum. [Note: This project is developing four inservice courses. One is for elementary school teachers; the others are for secondary school science, social science, and mathematics teachers. The first of these four courses is expected to be available late in 1987. The project will be completed by the end of summer 1978. The materials will be published by the International Council for Computers in Education.]

There is much basic research still to be done on lower-order versus higher-order skills. But the much larger research challenge is to gain the knowledge needed to incorporate computers into this whole area. A suggestion for the way technology is leading us is provided by the high tech/high touch ideas of Naisbitt (1982). It seems likely that high tech/high touch-oriented research will discover that increased use of computer-based technology will do two things:

1. Give an increased advantage to a person with high-order interdisciplinary-oriented skills who has been educated to make effective use of the technology.

2. Maintain or increase the importance of people-oriented skills.

Research on the role of intra- and inter- personal skills in an increasingly technological society may prove particularly rewarding. For example, the computer equity issue is often studied by looking at numbers of men and women taking various computer science courses or making certain types of uses of computers. The results are often used to argue for increasing the number of women in computer science courses. Suppose, instead, that a combination of computer application and interpersonal skills are much more broadly useful in a highly computerized society. We might then see an inequity running in the opposite direction and people beginning to encourage men to take appropriate coursework to improve their interpersonal skills.

WHOLE GROUP DISCUSSION: Have workshop participants share their ideas and experiences on the topic of the previous paragraph.

References

Adler, Mortimer J. *The Paideia Proposal: An Educational Manifesto.* New York: Macmillan, 1982.

Beyer, Barry K. "Improving Thinking Skills: Defining the Problem." *Phi Delta Kappan.* (March 1984): 486-490.

Beyer, Barry K. "Improving Thinking Skills: Practical Approaches." *Phi Delta Kappan.* (April 1984): 556-560.

Bloom, B.S., Krathwohl, D.R., and Masis, B.B. *Taxonomy of Educational Objectives.* New York: David McKay Co, 1956.

Boone, Randy. Ph.D. dissertation at the University of Oregon. December, 1985.

Carpenter, T.P., Corbitt, M.K., Kepner, H.S., Lindquist, M.M., and Reys, R.E. *Results from the Second Mathematics Assessment of Educational Progress.* Reston, Virginia: National Council of Teachers of Mathematics, 1981.

Davis, Bob. "Expert Systems Let a User Pick Stocks, Play Doctor More or Less Like a Pro." *The Wall Street Journal.* (June 10, 1985): 1.

ERIC. "Improving Student's Thinking Skills." *The Best of ERIC.* ERIC Clearinghouse on Educational Management, University of Oregon (December 1984).

Feigenbaum, Edward, and McCorduck, Pamula. *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World.* New York: New American Library, 1983.

Felt, Marilyn Clayton. *Improving Our Schools: Thirty-Three Studies that Inform Local Action.* Newton, Massachusetts: Education Development Center, 1985.

Ferres, G.W. *Training and Implementing Strategies Appropriate to the*

Introduction of Logo into Teacher's Curriculum and Instruction. Ph.D. dissertation, University of Oregon, 1983.

Fey, J.T. (Ed.). *Computers and Mathematics: The Impact in Secondary School Curriculum.* National Council of Teachers of Mathematics. 1984.

Fredericksen, Norman. "Implications of Cognitive Theory for Instruction in Problem Solving." *Review of Educational Research 54* (1984): 363-407.

Moursund, David et al. *Computer-Integrated Instruction Inservice Notebooks (for Elementary School and for Secondary School Science, Social Science, and Mathematics).* ICCE, at press.

Goodlad, J.I. *A Place Called School: Prospects for the Future.* New York: McGraw-Hill, 1983.

Naisbitt, John. *Megatrends: Ten New Directions Transforming Our Lives.* New York: Warner Books, 1982.

National Commission on Excellence in Education. *A Nation at Risk: The Imperative for Educational Reform.* Washington, D.C.: U.S. Printing Office, 1983.

NCTM. *An Agenda for Action: Recommendations for School Mathematics of the 1980s.* Reston, Virginia: National Council of Teachers of Mathematics, 1980.

NCTM. "The Impact of Computing Technology on School Mathematics: Report of an NCTM Conference." *Arithmetic Teacher.* (April 1985): 14-18.

Salton, G. *Automatic Information Organization and Retrieval.* New York: McGraw-Hill, 1968.

Salton, G., and McGill, M. *Introduction to Modern Information Retrieval.* New York: McGraw-Hill, 1983.

Sizer, Theodore. *Horace's Compromise: The Dilemma of the American High School.* Boston, Massachusetts: Houghton Mifflin, 1984.

Steidley, Karl. *An Empirical Determination of Instructional Objectives Relating to the Computer in the College-Level General Physics Sequence.* Ph.D. Dissertation, University of Oregon, 1983.

Toffler, Alvin. *The Third Wave.* New York: Bantam Books, 1980.

Travers, K.T., and McKnight, C.C. "Mathematics Achievement in U.S. Schools: Preliminary Findings From the Second IEA Mathematics Study." *Phi Delta Kappan.* (February 1985): 407-413.

Wetzel, Keith. Ph.D. dissertation at the University of Oregon. December, 1985.

Wheatley, Grayson. "Calculators in the Classroom: A Proposal for Curriculum Change." *Arithmetic Teacher.* (December, 1980).

Wilson, J.W., and Ticer, Scott. " Superchips: The New Frontier." *Business Week.* (June 10, 1985): 82-85.

Woodward, John. Ph.D. dissertation at the University of Oregon in Educational Psychology. June, 1985.

Chapter 3

The Future Of Computers in Education

Note: This chapter is a slightly modified version of Chapter 1.3 in the book *Long-Range Planning for Computers in Schools* written and published by Dave Moursund and Dick Ricketts. The book provides background information and general assistance to educational leaders working to develop five-year plans for instructional use of computers in their schools. It approaches long-range planning as a problem to be solved, and it contains a great deal of information useful in attacking this problem.

Chapter Summary

The first general-purpose electronic digital computer built in the United States became operational in December 1945. Computers became commercially available in the early 1950s and the market has expanded rapidly since then. This rapid expansion has been fueled by continued technological progress and rapid price-to-performance decreases.

Today's microcomputers outperform mainframe computers from 15 to 20 years ago, and we will likely be able to make a similar statement 15 to 20 years from now. Technological progress continues unabated.

So far, the educational impact of computers has been modest. However, we are now in a period of very rapid growth of computer availability in schools. We can now see some of the changes that computers may bring to education. It seems clear that eventually computers will be an everyday tool of students at all grade levels and throughout the curriculum. Both the content and the pedagogy of education will be profoundly affected.

This chapter is more technical than other parts of the book. It is suggested that the computer novice avoid getting bogged down in the details. Just be aware that substantial progress and rapid change will continue to occur in the computer field. Nine major recommendations are given at the end of this chapter.

Note: This chapter is based on a paper written by David Moursund for the summer 1986 Extensions of the Human Mind conference, sponsored by the University of Oregon. It has also been presented as a keynote address at a number of conferences.

Introduction

This chapter is divided into a number of sections. First is a general discussion of the basis for our predictions. The astute reader will notice that this chapter is **not** a summary of the literature (notice the lack of a bibliography at the end). However, there is considerable underlying logic and evidence to support the predictions. Most of the predictions are statements about current trends and are not quantitative in nature.

Next the chapter discusses some trends in hardware, software, networking, computer science, and overall applications of computers in instruction.

The chapter concludes with several recommendations. These are based on the overall prediction that computers will lead to profound changes in our educational system.

S-Shaped Growth Curve

General-purpose computers for use in instruction and research began to come into colleges and universities in the middle to late 1950s. The IBM 650, a first-generation (vacuum tube) machine, was typical of the commercially available computers in those days. If two 10-digit integers were placed in the right registers, it could add them in a thousandth of a second! When suitably programmed to calculate a square root of a 10-digit integer, it extracted the square root in an eighth of a second. Needless to say, students studying this machine were quite impressed by its "blinding speed." Such a machine served the entire research and instructional needs of major research universities such as the University of Wisconsin, Madison. Now, of course, millions of people own personal microcomputers with far more compute power. Some handheld calculator-computers have greater speed than the IBM 650. Today's super computers are between 100,000 and 1,000,000 times as fast as the IBM 650.

Interestingly, as soon as computers came into colleges and universities, people began to experiment with their use in high schools. Some high school students learned to write computer programs during the late 1950s. A few of these "high school hackers" went on to start computer companies that have had a profound impact on the computer industry. (Two excellent examples are provided by Apple and Microsoft.) However, widespread school use of computers had to wait for decreased prices and increased availability of the hardware.

Computers are an example of technology, and many other technologies have come into our society. For example, the telephone was invented by Alexander Graham Bell in 1876. But initially there were no telephone lines, central switching stations, or people trained in making effective use of telephones. Now, more than a century later, the telephone market is still continuing to expand.

After its invention, a technologically based product next undergoes the work needed to produce a marketable product and then is brought to market. Under free market conditions, product acceptance and distribution is characterized by an "S-shaped" growth curve.

```
                              ___________   <--- Saturated market,
                           __/                   Replacements only
                          /
                         /
                        /
                       /
                      /
                   __/
__________/     <--- Limited market, applications not realized
```

Initial growth in use and acceptance of a new product is slow because the product is not known to many people, the product is in short supply and may be quite expensive, the necessary infrastructure is not in place, etc. When television was first invented there were no television stations, sales outlets or repair shops.

Notice the period of rapid growth in the middle of the S-shaped growth curve. Both the number of years of rapid growth and the rate of rapid growth vary with the product, but eventually the market becomes saturated

The instructional use of computers in education seems to be following an S-shaped growth curve. In precollege education we seem to now be in the period of most rapid growth. Some authors suggest that the number of computers in precollege education is now doubling every 15 months. An educated guess is that there were about two million computers in precollege education in the United states in January 1987. This is a ratio of about one computer per 23 students.

However, the reader should be aware that computers in schools may not continue to follow the S-shaped growth curve model. The education market is not like the general consumer-based market. Funding for education is very tight, so computers-in-education may not follow a standard free market growth pattern. Indeed, in most school systems the majority of computer hardware has been purchased using non-recurring funds, such as those from various federal entitlement programs. Large cuts in these programs are possible, and there are other pressing demands on these funds.

Four Eras

A number of writers have examined and characterized major trends in human societies. They have identified four major ages: Hunter-Gatherer, Agricultural, Industrial, and Information. Until about 10,000 years ago, all societies were Hunter-Gatherer in nature. Then the Agriculture Age emerged in various parts of the world. About two hundred years ago the Agricultural Age began to give way to the Industrial Age in some parts of the world . Finally, the Information Age began to emerge about 30 years ago in some industrialized societies. In the United States, the official advent of the Information Age is often listed as 1956.

The label "Information Age" is misleading. What has happened in the United States since the end of World War II has been a massive shift from a manufacturing economy into a service economy. For example, the number of fast food stores has grown very rapidly, as has the number of clerks working in these stores. These are certainly not high-tech, information-oriented jobs. Most of the service jobs pay far less than manufacturing jobs. A combination of a decrease in manufacturing (blue collar) jobs and an increase in service jobs is having a significant impact on the U.S. economy.

But information-related technology is certainly an important part of our Information Age. Who can help but be aware of the rapid growth of IBM, whose 1986 sales were about fifty billion dollars ($50,000,000,000)? And computer-related technology has transformed many industries. For example, our telephone system is much better than it was when we were children, due to computerized switching and other computer equipment, satellites, fiber optics and so on.

Computers and computer-related technology are playing two ever-increasing roles in our Information Age. First, computers are a productivity aid in many situations. (That is, they are an aid to solving the problem of helping a person be a more productive worker.) These situations range from a writer using a word processor to a librarian searching computerized data banks to a computerized robot helping assemble automobiles. Second, computers are being built into many devices such as automobiles, microwave ovens, and television sets. Sometimes computer technology makes possible entirely new devices such as electronic digital wrist watches and handheld electronic calculators. These two general categories of computer use--**productivity** and **products**-- are now well established and will surely continue. They serve as a solid foundation for predictions about the future of computers in education.

Computer Hardware Trends

To a large extent the computer field has been driven by hardware progress, and this progress is showing no signs of slowing. We will examine these trends and make a couple of general predictions. You should notice that these predictions are made quite independently of whether computers become of increased importance in education. That is, the education market is not yet large enough to be a major driving force in the development of hardware. We have estimated that as of January 1987 there were about two million microcomputers being used for instructional purposes in precollege education in the United States. The entire cost of all of these computers is less than five percent of IBM's sales for the year 1986!

The ENIAC, which became operational late in 1945, was the first general-purpose electronic digital computer produced in the United States. It contained 18,000 vacuum tubes at the very heart of its computational circuitry.

The transistor was invented in 1947, and one can think of a transistor as being roughly equivalent to a vacuum tube. It took about 10 years before transistorized computers began to become available. At that time transistors cost about the same as vacuum tubes but used less power and were more reliable.

In much less than another 10 years the integrated circuit was in mass production. The IBM 360 series of computers that came out in the mid-1960s made use of integrated circuitry and these were called third-generation machines. A historical note seems appropriate. When Dave Moursund began teaching at the University of Oregon in the fall of 1967, an IBM 360 model 50 mainframe computer had just been installed. The UO was quite proud of this new machine with its high speed 32-bit CPU, 256K-byte memory, multiple tape drives, and two five-megabyte disk drives. Later, at a cost of about a quarter million dollars, eight 27-megabyte disk drives were added to the system. The total system cost nearly a million 1967 dollars.

(Technical Note: The central processing unit runs the computer and is usually called a CPU. Some people call it the brain of the computer. A bit is the smallest unit of informtion. One can think of it as an one of many on/off switches in a computer. A byte is a unit of eight bits. A computer can store any letter or digit in one byte. In the computer field, a K means 1,024 of something, and "mega" means 1,048,576 of something.)

In terms of CPU capability, primary storage and secondary storage, the cost of computers has come down by a factor of perhaps 500 during the past 20 years. Many people now own personal microcomputers with more primary storage and raw compute power than the IBM 360 model 50. There has also been substantial progress in software, a fact discussed later in this chapter.

The trend of packing more and more circuitry into a small chip has continued unabated. Chips containing the equivalent of more than two million transistors (these megabit chips can store 128K bytes of data) are now in mass production. Researchers seem confident that a further gain by a factor of 32 or 64 will be achieved by the year 2000. The next steps seems clear. Researchers in a Texas Instruments lab produced a four-megabit (that is, 512K-byte) memory chip in 1985. IBM announced in 1987 that it has learned to produce such a chip in the same production facility it is currently using to mass produce the megabit chip. Such a product should be in mass production by a number of companies before 1990. Reseachers at Nippon Telegraph and Telephone announced in January 1987 that they had produced a 16-megabit (that is, a two-megabyte) chip that should be commercially available in1990 or 1991.

The hardware progress led to single chip 32-bit CPUs in 1981, second-generation single chip 32-bit CPUs in 1985, and a variety of still faster 32-bit CPUs since then. The Intel 80386 and the Motorola 68020 32-bit CPU chips introduced in 1986 rival the CPUs of medium-priced mainframe computers being produced at the same time. Considerable progress in commercially available parallel processors, involving the interconnection of multiple CPUs and primary memory, has occurred during 1985-86. The "neural-network" type of computer is just now starting to become commercially available. This type of hardware design interconnects a large number of processing units, in much the same way neurons are interconnected in a brain. Such hardware can solve certain types of problems very rapidly.

Progress toward building faster and more powerful mainframe computer systems continues. Increased speed is coming from a combination of faster circuitry and use of multiple CPUs (parallel processing). Recent (1987) breakthroughs in super conductivity may allow the production of circuits that are 100 to 1,000 times as fast as the current fastest circuits.

There is a strong trend toward building computers with very large memories. In 1986 several companies announced plans to build computers with a billion bytes of primary storage. Such machines will be particularly effective in solving certain types of problems that involve a large amount of data. Weather forecasting is an example.

Computer hardware has made so much progress that it is now possible to design special circuitry and build special-purpose computers for a particular application. At Carnegie-Mellon University, for example, a special-purpose chess playing computer has been designed and built. This machine employs 64 special-purpose processors (one for each square on a chess board) in addition to a general-purpose processor which runs the master program and coordinates the overall efforts. The system can examine 175,000 possible chess moves per second. It recently won the North American computer chess title. Only a few hundred humans in the entire world play chess better than this computer.

The trend toward greater packing density in computer circuitry has made portable microcomputers possible. Portable microcomputers are now available ranging in capability from not much more than a sophisticated pocket calculator up to machines fully as powerful as medium-priced microcomputer systems. Portables range in price from about $500 up to about $8,000. One machine near the lower end of the price scale, from Radio Shack, has a 16-line display, a built-in word processor and several other pieces of software, and a modem.

Display screens and batteries remain major problems for portable microcomputers. Slow progress is occurring with display screens. Circuitry using less power is being developed, and batteries are very slowly being improved. It seems clear that the cost of portable microcomputers will slowly decline, even as their computing power gradually increases. The market for medium-capability portable microcomputers is still modest in size and prices are declining only slowly.

An interesting use of portable microcomputers is taking notes in a class or at a meeting. A few college students now use such computers to take notes during class lectures or in their library research. Educational research has not yet addressed the issue of whether a student gains a significant advantage by using a computer to take notes.

To summarize, we can predict with considerable confidence that computer CPU and primary memory hardware will become more and more powerful and that the real costs of a given amount of compute power will continue to decline quite rapidly. (It appears that the price to performance ration of computers has been improving by a factor of about 10 every seven years for the past 28 years.) This means that any hardware one purchases now will be surpassed in a few years. That difficulty, which has already lasted for four decades, will continue for at least a couple more.

Besides raw compute power, rapid progress continues in secondary storage devices and in input/output devices. A brief discussion of each of these areas follows.

Secondary storage is making very rapid progress in both magnetic storage devices and optical storage (i.e., laser) devices. Floppy disk technology continues to progress. In1986 a floppy disk system with a storage capacity of over five megabytes was brought to market. Hard disk systems are rapidly decreasing in price and increasing in capacity. The cost of a 20 or 30 megabyte hard disk system for a microcomputer is now well under a $1,000. Some brands of these hard disk systems now retail for about $500. Much larger capacity hard disk systems are now quite common on microcomputers. For example, one can now purchase 100 to 150 megabyte hard disk systems for a microcomputer.

Optical storage technology is now a commercial reality. Videodiscs were initially used to store movies and television programs. One side of a disc can store 30 minutes of television. This amount of storage can hold up to 54,000 individual pictures, such as pictures of art objects or pictures taken through a microscope. The potential educational value of such videodiscs is obvious. A single disk can contain photographs of every important painting in the world's major art museums.

Many people own hi-fi systems that play audio compact digital discs. A somewhat modified version of these discs (called a compact disc-read only memory, or CD ROM) can store computer data. The 550 megabyte CD ROM system (the disc itself is only 12 cm--about 4.72 inches--in diameter) is already finding many commercial applications. For example, poison control centers using laser disc systems find they can do better searches (using boolean logic) and can access information much faster from such a system then they can from microfiche. *Books in Print* is now available on CD ROM, along with order information to aid in purchasing any listed book. The ERIC system is now available on CD ROM.

The technology for audio disks and for data disks can be combined in a single system. Such systems will soon become commercially available. Current predictions are that such combined systems (providing hi-fi, information retrieval, and a built-in computer) will prove quite popular and will find wide consumer acceptance. This will greatly increase the number of general-purpose computers available in homes. Predictions are that such systems will begin to come to market in 1988 and will have a major impact during the subsequent five years.

Write once, read many-times (WORM) optical disc technology is now on the market. One application for such technology is for preparation of large data banks that will be distributed (for example, by mail) to a number of sites.

The write/erase technology for optical disks has been demonstrated and is headed toward the commercial marketplace. Perhaps by 1990 systems of this sort with a billion bytes of storage on one disk will be available. All of this optical storage technology will help provide increased access to larger and larger data banks.

In the input/output arena, there are two driving forces. One is to improve the human-machine interface for computer users. This can be done, for example, by suitable use of voice or sound output, voice input, graphic displays, mouse, glove, touch screen, etc. The second is to reduce the cost and improve the quality of "conventional" input devices and hardcopy output display devices.

The voice input problem has represented a considerable challenge to researchers in computer science. Significant progress has occurred. Disconnected speech systems (requiring a very short break between words) have been commercially available for quite some time. The connected speech problem, handling the way people usually talk, has proven to be a major challenge to researchers in artificial intelligence. It is now clear that eventually there will be systems that can process normal speech with considerable accuracy. While business applications are clear, imagine the applications to teaching reading and writing to young children! The teaching of reading and writing could merge together and might well occur a couple of years earlier than the current average.

The past few years have seen a strong trend toward high-resolution graphics displays (often with color) for microcomputer systems. The more expensive systems frequently used in science and engineering often have 1,000 by 1,000 bit-mapped displays. Top-of-the-line displays have a resolution of 4,000 by 4,000. The computer can turn on any of of 16,000,000 dots on its monitor screen! The need for large memories becomes clear as one considers bit-mapped graphics. It requires 16,000,000 bytes of storage to store one full-screen color display.

Humans have considerable skill at recognizing visual patterns and in processing information presented visually. Computer-generated motion graphics on high-resolution display devices adds a new dimension to the human-machine interface. It seems evident that uses of high-resolution graphics will continue to expand.

Computer hardcopy output devices (impact printers and laser printers) continue to make rapid progress. In 1975 a printer that cost about $3,500 could print10 characters per second, upper case only. Now one can purchase impact printers that are more than 10 times as fast, produce much better images, handle graphics, and cost less than a 10th as much when the price is adjusted for inflation. Note also that several color impact printers are now on the market. While some additional progress can be expected in impact printers, it will not be nearly as rapid as in the past decade.

Laser printers are now coming to the fore. In essence, a laser printer uses a xerographic printing process. The past couple of years have seen rapid decreases in price and a number of new laser printer products. A period of rapid progress is occurring, and it should continue for quite a few years. Substantial decreases in prices can be expected.

The most common laser printer has a resolution of 300 dots per inch. This is not quite as good as typeset materials, but is about four times the resolution of typical dot matrix impact printers. A 1,200-dots-per-inch laser printer is also available. Its quality is comparable to typeset materials. Laser printers and their associated computer systems have spawned the desktop publishing industry.

Software Trends

Software progress over the past 40 years has been steady, but certainly not as spectacular as hardware progress. However, software progress will continue steadily for many centuries to come, long after hardware progress has slowed. We will briefly discuss three aspects of software:

- Programming languages and operating systems
- Applications software
- Programming systems

Rapid progress in programming languages occurred quite early in the history of computers. From the early use of machine language it was a relatively quick step to assembly language. Higher-level languages such as FORTRAN (developed in 1953-1957) and COBOL represented major progress. (Higher-level languages are easier than lower-level languages for humans to work with.)

Many other high-level programming languages were developed to fit the needs of specific groups of users. One interesting aspect of this is how long a language lasts once it gains a significant base of users. For example, FORTRAN is still alive and well, and the current versions are much better than the early versions. FORTRAN is still commonly used by scientists and engineers, and there are literally billions of dollars worth of software written in FORTRAN.

Since the late 1960s there has been a strong trend toward structured programming languages such as Pascal and C. An individual programmer using these languages is not much more productive than one using FORTRAN or COBOL. But teams of programmers can much more easily work together if they use the newer languages, and the resulting products are more maintainable. Errors can be fixed (debugged) with *relative* ease. In terms of maintainability and modifiability, structured programming languages have led to progress by a factor of perhaps three to five over use of unstructured programming languages.

The quest for still better programming languages continues. One result is Ada, which has received strong support from the U.S. military system. However, many professional programmers suggest that Ada represents a great leap backwards. The attempt to develop a language that will be "all things to all people" has not met with great success. This suggests that it is difficult to develop a general-purpose programming language that is clearly superior to existing languages.

Our guess is that computer scientists will not make very rapid progress in developing still better general purpose programming languages in the next decade. One reason for this is that each discipline (that is, each major area of problem solving) has its own vocabulary, notation, data representations, and data structures. A programming language specifically designed for use in a particular problem solving area will tend, in that area, to be superior to a general purpose programming language designed for use across all disciplines.

The initial progress in developing better operating systems paralleled progress in developing better programming languages. We now take such things as timeshared operating systems on a mainframe and mouse- based manipulation of files on a microcomputer for granted. But these represented major progress when they first were developed. In some ways we have witnessed a merging of programming languages and operating systems, such as programming in C in a Unix operating system environment.

Parallel processing and neural-network systems provide a major challenge to computer scientists who specialize in operating systems. The goal is to have programming language and operating system aids to help the programmer effectively use these types of hardware systems. Progress has been, a[illegible] ...kely to be, slow.

Next we look at applications software. Here we sometimes forget the progress that had already occurred 25 years ago. By the [illegible] 1960s there were substantial computer libraries of scientific subroutines and statistical packages. Now, of course, such programs run on microcomputers and are a little more user friendly.

There seem to be two clear ideas in the computer applications area. First, any application currently running on a mainframe computer will eventually be available on a microcomputer. User friendliness will be improved to increase the mass marketability of the product. Here it is worth mentioning IBM's 1986 announcement that it has produced an IBM 360/370 CPU chip. This suggests that eventually much of the IBM software library will become available to microcomputer users. This library contains many billions of dollars worth of software.

Second, artificial intelligence is now producing a large number of aids to problem solving. Artificially intelligent applications software is increasing the range of problems a computer can handle. Many hundreds of such products have come to market, and more will follow. This trend will surely continue through the work of artificial intelligence researchers and as the needed compute power becomes more cheaply available. The challenge to education is immense. *If a computer can solve or help solve a particular category of problems, what should students be learning about how to solve this type of problem?*

Voice input, which was discussed earlier in this chapter, provides a good example of the educational challenge of artificial intelligence. Early voice input systems required the power of a mainframe computer system and took a minute or more to decipher a short utterance. Now there has been considerable progress in the theory of voice input, and computers have become both faster and cheaper. Consequently, voice input systems will gradually grow in use during the next decade. However, a connected speech system will remain elusive for the next several years.

Networking

As mentioned earlier, one characterization of the Information Age is better access to more information. The worldwide telecommunications system is growing very rapidly, aided by satellites, microwave systems, and fiber optics. Local area networks (LANs) are now common, and this type of computer interconnection is in a period of very rapid growth. Such systems are currently being installed in the United States at the rate of perhaps 50,000 to 75,000 systems per year. The new (1987) series of IBM microcomputers was designed for easy networking.

It is hard to appreciate the potential here. Optic fibers will eventually replace the copper wires now connecting our home telephones. The capacity of a pair of optic fibers, each as thin as a human hair, is thousands of times the capacity of a pair of copper wires. Fiber optics can support high-quality interactive color television. Thus, we will eventually have picturephones!

Rapid progress is still occurring in fiber optic technology. For example, researchers are beginning to produce optical integrated circuits. Such researchers talk about building optical computers that may be a thousand times as fast as today's computers.

But that is many years in the future. More down to earth is the increasing carrying capacity of optic fibers. Recent research results suggest the possibility of a single optic fiber having the potential to carry 10 million simultaneous two-way telephone conversations. This capacity would allow a full-length color movie (appropriately digitized, of course) to be transmitted in less than a second.

A combination of improved networking and improved large-scale database storage devices will have a profound impact on libraries and the publishing industry. We mentioned previously the 12cm CD ROM with a capacity of 550 million bytes. A single disc stores the equivalent of 500 very thick novels--or more than the number of books a typical student studies during four years of college! A handful of these discs can store the equivalent of a typical school library.

We must realize that libraries and the publishing industry are very large, relatively slow-moving institutions. They have started to adjust to computer technology; this change will be slow but steady. Already we can see significant changes in research libraries and in the storage/retrieval of literature to support research. Also, many libraries have replaced their card catalogs with computerized sytems.

Computer Science

Computer science is now a well-established discipline. The Association for Computing Machinery first developed recommendations for the content of an undergraduate computer science degree in 1968. There are now hundreds of computer science departments at the college and university level. There are several competing national recommendations on curriculum and there are textbooks to support these recommendations. A very large number of college students are currently majoring in computer science. Interestingly, however, such enrollment in the United States has peaked and is now declining. Perhaps students are seeing that computer science is both a hard major and no guarantee of a high-paying job. It appears that the market for bachelors degrees in computer science has been saturated.

During the past 30 years we have seen computer science course content filter down from the graduate level to the undergraduate level and even into the precollege level. List processing and recursion in Logo, a language designed for elementary school students, certainly illustrate this point. The Logo language draws heavily from a programming language developed for use by artificial intelligence researchers and was formerly the province of graduate students in computer science. As another example, the high school Advanced Placement course in computer science contains many topics that used to be studied at the advanced undergraduate level.

While the discipline of computer science is beginning to mature, it seems evident that rapid change will continue. An excellent example is provided by the Prolog programming language. A few years ago it was primarily of interest to graduate students and researchers specializing in artificial intelligence. Now it may be encountered in a freshman course for computer science majors. It is even being used with grade school students in a research project in England.

College and university computer science departments face an interesting challenge. As computer science matures, it becomes a more rigorous, mathematically oriented subject. It is a discipline that challenges the intellectual capabilities of people who are good at algorithmically oriented problem solving. But a major goal of computer science is to make computers easier to use, and rapid progress is occurring. Today's graphic artists, perhaps with no formal training in computer science, routinely solve problems that were at the frontiers of computer graphics only 20 years ago.

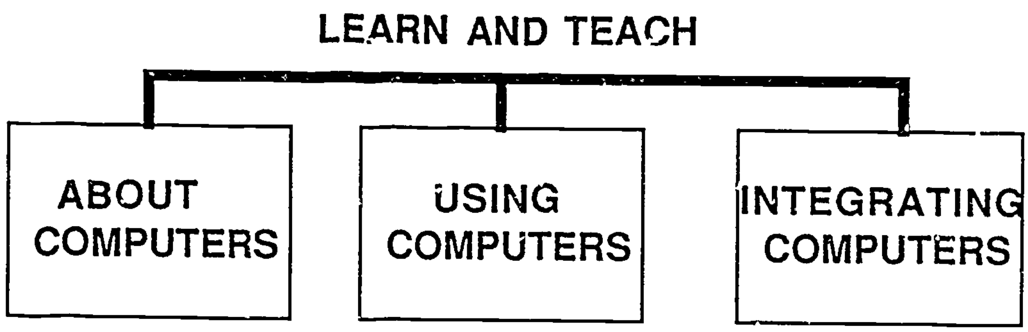
The question is, who needs to study computer science, and to what extent? At the current time there is no consensus. It is evident that computer science will continue to be a major field of study in colleges and universities. However, it seems likely that the serious study of this discipline will decline at the precollege level. Certainly the serious study of computer science in high schools is no longer a growth area. Total enrollment in such courses is declining in the United States.

Computers in Instruction Trends

Here we will take two approaches, treating each somewhat briefly. In the first approach we will examine an overview of computers in instruction. There (as discussed in Chapter 2 of this Independent Study course book) the total field is divided into teaching/learning about, using, and integrating computers. In the second approach we will look at instructionally oriented software, instructional support materials, and teacher training. There are clear trends in all of these.

We begin by examining our diagram which divides the overall field of computers in instruction into three major parts.

COMPUTERS IN INSTRUCTION



Learn and Teach About Computers

In the "About Computers" category we include the discipline of computer and information science, of which computer programming is a subfield. The key issue is what we want students to know about computers. We want to dispel the magic, and we want all students to have a mental model of a computer as a machine that can follow a detailed step-by-step set of directions that have been developed by humans. Typically the step-by-step set of directions has been designed to solve or help solve some category of problems. We want all students to have a reasonable level of knowledge about roles of computers in problem solving.

The heart of the matter seems to be the idea of effective procedure--developing and representing procedures designed to accomplish specified tasks. We want students to have some idea about what is involved in a carefully developed specification of a procedure (data representation, data structures and control structures). We want students to understand how difficult it is to develop procedures, and that there are many disciplines (such as the arts and humanities) which now have few computerizable procedures.

We want students to understand roles of computers in problem solving in all disciplines. Most educational leaders suggest that we don't do very well in non-computer-based problem solving. Thus, we tend to confuse computer-based problem solving with the more general issue of problem solving. This general issue will continue to beguile educators indefinitely.

It is clear that some form of computer programming can be taught at almost any grade level. However, we lack solid evidence that the teaching of computer programming helps improve general problem solving skills. That is, problem solving skills learned in a computer programming course do not transfer automatically to solving problems in other courses or disciplines.

Also, we now know that if we want students to gain increased skill in writing programs to solve problems, we have to place more emphasis specifically on problem solving. Many people who are currently teaching computer programming courses are not particularly qualified or trained in the knowledge and skills needed to significantly increase the emphasis on problem solving.

Our conclusion is that the above types of issues concerning teaching computer science and computer programming at the precollege level will remain unresolved during the next decade. Individual school districts will decide what seems best for their students. Eventually there will be sufficient research to answer some of the questions.

Learn and Teach Using Computers

Learning and teaching using computers is often called computer-assisted instruction (CAI) or computer assisted learning (CAL). Much of the commercially available CAL material also contains a record keeping, diagnostic testing and prescriptive system--that is, a computer-managed instruction system.

It is likely that we are at the start of a steady and significant rate of growth in use of CAL. Six factors combine to support this prediction. First, research evidence to support the effectiveness of CAL is quite strong and continues to grow. Second, declining hardware costs are making CAL more and more cost effective. Third, there is an overall educational trend in the United States toward individualization of instruction, and CAL supports this trend. Fourth, CAL can increase productivity of students and teachers. Fifth, the amount of CAL material is now of significant size and growing quite rapidly. Sixth, CAI is improvable; newer software will usually mean better software.

Several commercially viable companies offer a broad and rapidly growing collection of materials. The Computer Curriculum Corporation, which was started by Stanford University professor Patrick Suppes in the late 1960s, now (1986) has about 3,500 hours of materials on the market. One standard use of quite a bit of this material is for routine drill and practice. For example, there are materials to support 10 minutes of arithmetic drill daily throughout grade school. The materials from this one company cover a significant part of the entire precollege curriculum.

We predict that CAL will gradually become a significant factor in the education of most students. We wouldn't be surprised to find that 15-20 years from now more than half of all precollege instruction in this country being delivered in a CAL mode. This suggests a major change in the role of teachers, changing from delivery of instruction to facilitation of learning.

Learn and Teach Integrating Computers

The integration of computer-as-tool into the curriculum is now well started. Many school districts have set goals of having all students learn to use a word processor, database system and graphics package. The key idea is that such tools increase user productivity--the ability to solve problems. The evidence for increased productivity in the work place is overwhelming and is a core idea of the Information Age. This evidence seems to be motivating schools to change their curricula.

We know, of course, that it is difficult to make major changes in school curricula in a short time. Over the next two decades, those who understand and appreciate the potential of computer-as-tool will mount a major challenge to the inertia of our school system. It seems clear that eventually computer-as-tool will be thoroughly integrated into every academic discipline. This will require major changes both in curriculum content and in testing. Students will need access to computers during testing, and this will be a major challenge to the testing industry.

The Growing Software Infrastructure

Our second approach to predicting the future of computers in instruction is to look at instructionally oriented software, instructional support materials, and teacher training. *The Educational Software Selector*, published by EPIE, lists nearly 8,000 pieces of software. It is clear that the average quality of educational software is growing. The market is now large enough to support a number of companies that specialize in educational software. The competition, as well as excellent leadership in some of these companies, is contributing to improvements in educational software.

Perhaps especially noteworthy is a trend toward development of tools and CAL materials to support longer units or even full courses of study. For example, several companies offer a year-long high school geometry course in a CAL mode. *The Geometric Supposer*, a series of four major pieces of software published by Sunburst Communications, is an excellent tool for use throughout a high school geometry course. Such software provides solid evidence that eventually most of the precollege curriculum will be available in a CAL mode. This will increase options available to schools and students, and thus it will bring an increased element of competition into our educational delivery system.

Videodisc-based CAL is gradually growing in importance. A clear trend has been established, with leadership coming from business, industry, and military education. Some excellent materials are now available for use in the precollege curriculum. We predict slow but steady growth in videodisc-based CAL in precollege education. That is, we are still on the lower leg of the S-shaped growth curve for this type of CAL.

Some of the videodisc-based and other CAL materials are being designed to maintain the teacher at the center of the instructional delivery system. The teacher controls the videodisc system or a single computer being used for a computer simulation. The teacher facilitates class discussion and other activities Relatively few teachers have been trained in such use of technology in education, so this suggests an area where extensive staff development will be needed.

Instructional support materials include books, films, reproducibles, and so on. Conventional publishers understand the production of these types of materials. An indication of how much material is available is provided by the 1985 state of Texas textbook adoption for junior high school computer literacy. There were 13 books, all copyrighted 1985 or 1986, submitted for their consideration. The computer literacy market is large enough to motivate a number of companies to produce products specifically for it.

The past few years have also seen the publication of a large number of computer-related books to support teacher training. Indeed, this market has been saturated in the past two years (1985-86), and a number of companies have cancelled some planned products. In total, it seems clear that publishers are well aware of the potential for sales of computer-related instructional support materials. Indeed, although there is still plenty of room for high-quality materials, it seems the quantify of materials already available is quite large. This is now a maturing segment of the publishing industry.

The teacher training problem is quite large. In the United States there are well over two million inservice precollege teachers, and their average level of computer knowledge and experience is still quite low. Most school districts recognize this problem, and the response of school districts throughout the country has been good. A number of districts have provided all of their teachers with introductory training. The number of certificate programs and master's degree programs in computers-in-education has rapidly expanded in the past three years.

Two things seem clear. First, the average level of teacher computer literacy will continue to grow steadily for many years. Second, this average level of teacher computer literacy will continue to be quite a bit lower than what computer education leaders might desire. Very few current teachers have grown up with computers. We still have very few students entering college who began using computers while they were in elementary school. However, a significant fraction of students now entering college have used computers for several years during their high school education.

Conclusions and Recommendations

Many of the trends discussed in this chapter seem quite clear. The hardware price-to-performance ratio for computers will continue to improve quite rapidly. Hardware will be networked. More and better software will become available. Computers will solve or help solve an increasing range of problems. Artificial intelligence will grow in importance and in use. In summary, our access to information and aids to processing this information will increase many fold in the years to come. Computer use in government, business, industry and education will continue to grow quite rapidly.

We believe the educational implications are profound. Nine general recommendations are given below. Their full implementation would lead to major changes in our instructional system.

Recommendation 1. Computer-assisted learning should be viewed as an aid to student-learning productivity. There should be considerably increased emphasis on CAL to make broader educational opportunities available to students, to facilitate more individualization of instruction, and to increase learning.

Recommendation 2. Computer-as-tool should be viewed as an aid to students' productivity in making use of their education (that is, in solving problems), while gaining more education, and eventually on the job. All instruction at all levels should take into consideration computers as an aid to problem solving and computers as a source of problems. The use of computer-as-tool should be integrated throughout the curriculum. Curriculum content and testing should be modified adequately to reflect computer-as-tool.

Recommendation 3. Computer-as-tool should be viewed as an aid to teacher productivity. All teachers should have easy access to computers to aid them on the job as well as in their own learning. Every teacher should have access to a personal computer at work and at home. Almost every classroom should have a computer with large display screens or a projector to allow computer-aided interaction between teacher and class.

Recommendation 4. Our preservice and inservice teacher education program must be modified to meet the computer challenge. All teachers have an increasing need for both general computer literacy and relatively deep knowledge on uses of computers within their own specific subject areas. School systems should place much more emphasis on providing educators with such learning opportunities and facilitate their taking advantage of such opportunities.

Recommendation 5. All preservice and inservice teachers should be given appropriate opportunities and encouragement to improve their abilities to function well in this changing environment. Computers will change the role of teachers. There will be less demand on teachers to be the source of information and the delivery device. There will be increased demand to be a facilitator--a role model who aids students in acquiring increased interpersonal skills and higher-order thinking and communication skills.

Recommendation 6. All schools should provide good access to computer-based information systems. All students should be given instruction in use of such systems and should make regular use of these systems throughout their schooling. The total accumulated knowledge of the human race is growing rapidly. Learning to access and make appropriate use of this collected information is at the core of education.

Recommendation 7. The concept of effective procedure (including the creation and representation of procedures, and algorithmic thinking) is among the most important academic ideas of our century. All students should learn this concept and its impact on various aspects of what it means to be educated for life in our society. Students should learn enough of the general capabilities, limitations and underlying nature of computers so that the magic of computers is dispelled.

Recommendation 8. Most real-world problems are interdisciplinary in nature. Schools should place increased emphasis on cross-fertilization among disciplines, on applications of one discipline to the study of a second, and on solving problems by 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 carrying it out.

Recommendation 9. Computers are changing our world view, our metaphors, our ways of handling everyday issues and problems. We should be aware of ways computers are changing our world and not lose sight of important underlying values as we adapt.

Chapter 4

Computers and Problem Solving: A Workshop for Educators

Note: Chapter 4 contains the entire contents of the booklet *Computers and Problem Solving: A Workshop for Educators*, written by David Moursund and published by the International Council for Computers in Education in 1986. A few corrections and additions have been made to the original book.

Contents

4.1 Preface
What is a Problem?

4.2 Key Ideas in Problem Solving

4.3 Roles of Computers

4.4 Accumulated Knowledge of Humans

4.5 Effective Procedures

4.6 Conclusions and Recommendations

4.7 Appendix on Active Listening
References

4.1

Preface

Over the past few years I have presented a large numb 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

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.

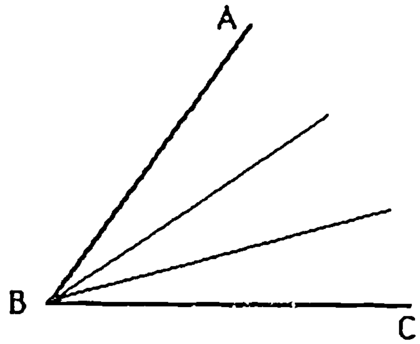
The issue of Ownership is particularly perplexing to educators. They recognize that Ownership--deep interest and involvement with a situation--often contributes to deep and lasting learning and intellectual growth. Thus, teachers often expend considerable effort to create situations that will get their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



Given 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 reduc... use of pirated software by two-thi... 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 b 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 metnod 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 tnrough 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?

4.2

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?

Debrief. 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 currciulum materials that are designed to help teach such techniques.

[Response 17 was discussed briefly earlier in this Computers and Problem Solving: An Independent Study Course *book. It is 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 expecially 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.

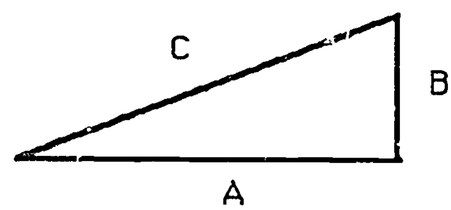
1. Understand the problem.

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.

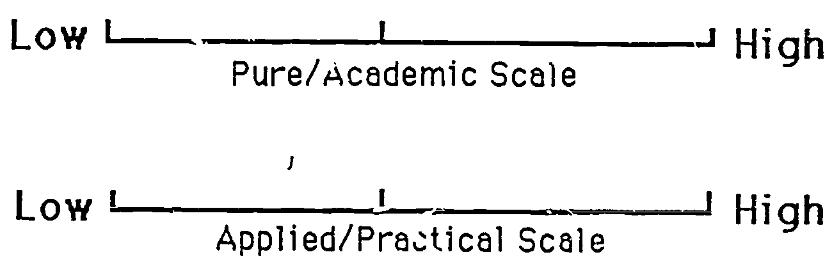
Exercise. Please bring to mind a problem that you have recently encountered and solved. It can be the one you thought of at the beginning of the workshop, or it can be a new problem. Do a quick mental check to make sure that it satisfies our formal definition of a problem. Then carefully think through 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 importan[illegible] I acknowledge their idea and indicate that is why so many b[illegible] 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 school work--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.

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. O 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 consicious 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 nonschool 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.

4.3

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 nelping 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:

$$H_2SO_4 + 2NaOH \rightarrow Na_2SO_4 + 2H_2O$$

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 explaination 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 individu. 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 a' . tudents 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.

Debrief in triads. 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--subproblems 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 (i.e., Section 4.5 of the Independent Study book) 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 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 pla· 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 seem to 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.

4.4

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 w 's/places in which information is stored, and how these are . affected by c nputers.

I have built a simple model which 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--they 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 nonschool settings. Some machines 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. 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 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.

4.5

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, nonthinking 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, 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, nonthinking 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 artifically 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 it 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 anwer 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.

4.6

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.

Fewest Most

1---2---3---4---5---6---7---8---9---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 noncomputer 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 noncomputer-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 natonal 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 multidigit 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

Thoughts on Computer Programming

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 comparisions 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 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 exibited 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

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.

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 symbology 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. The representational language 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?

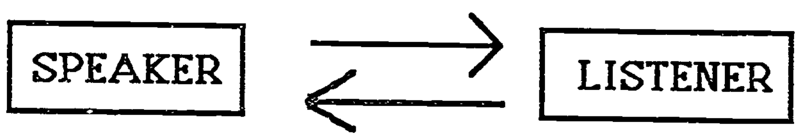
It would be nice if the clear and overwhelming answer to the questions were "Yes!" Unfortunately, there is little research to suggest this is the case. Moreover, resources devoted to creating a Logo environment and student time spent working in that environment could be used in other ways. Perhaps the 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.

Appendix

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 time keeper. 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.

References

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

Beyer, Barry K. "Improving Thinking Skills: Defining the Problem." *Phi Delta Kappan* (March 1984): 486-490.

Beyer, Barry K. "Improving Thinking Skills: Practical Approaches." *Phi Delta Kappan* (April 1984): 556-560.

de Bono, Edward. *New Think.* Avon Books, 1971.

de Bono, Edward. *Lateral Thinking: Creativity Step by Step.* Harper Colophan Books, Harper and Row, 1973.

ERIC. "Improving Students' Thinking Skills." *The Best of ERIC.* ERIC Clearinghouse on Educational Management, University of Oregon (December 1984).

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

Fredericksen, Norman. "Implications of Cognitive Theory for Instruction in Problem Solving." *Review of Educational Research* 54 (1984): 363-407.

Gardner, Howard. *Frames of Mind: The Theory of Multiple Intelligences.* Basic Books, 1983.

Jastrow, Robert. *The Enchanted Loom.* Simon and Schuster, 1981.

Kulik, J.A., Bangert, R.L. and Williams, G.W. "Effects of Computer-Based Teaching on Secondary School Students." *Journal of Educational*

Psychology. 1983, 75, 19-26.

Mayer, Richard E. *Thinking and Problem Solving: An Introduction to Human Cognition and Learning*. 1977.

Moursund, David. *High Tech/High Touch: A Computer Education Leadership Development Workshop*. International Council for Computers in Education. Third edition, April 1986.

Naisbitt, John. *Megatrends: Ten New Directions Transforming Our Lives*. Warner Books, 1984

Papert, Seymour. *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books, 1980.

Polya, George. *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press, 1957.

Rubinstein, Moshe F. *Patterns of Problem Solving*. Prentice-Hall, 1975.

Rubinstein, Moshe F. *Tools for Thinking and Problem Solving*. Prentice-Hall, 1986.

Toffler, Alvin. *The Third Wave*. Bantam Books, 1980.

Torrance, J. Pansey. Torrance is the author of a large amount of material on creative problem solving. His address is:

Georgia Studies of Creative Behavior
Dept. of Educational Psychology
The University of Georgia
Athens, GA 30602

Tuma, David and Reif, Frederick, eds. *Problem Solving and Education--Issues in Teaching and Research*. Lawrence Erlbaum, Hillsdale, N.J., 1980.

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

Chapter 5

Sample Materials from a National Science Foundation Project on Effective Inservice for Integrating Computer-as-Tool into the Curriculum

This chapter consists of the Preface and one chapter from *Computer-Integrated Instruction: Mathematics*, a book being developed through a National Science Foundation project directed by David Moursund. The materials in this chapter were developed by Seymour Hanfling, Tsuki Harriman, Judi Mathis, Jim McCauley and David Moursund. A few modifications have been made so that the materials fit in better with the *Computers and Problem Solving: An Independent Study Course* format and ideas.

The NSF project will be developing four books. The first (on mathematics inservice) should be available late in 1987, and the rest should be available before the end of 1988. They will be published by the International Council for Computers in Education. Each book will be available both in hard copy and on a Macintosh disk (in MacWrite format).

Each of the books will contain the same preface, overview of computers in education, overview of effective inservice education, etc. But each book will contain detailed information on eight inservice sessions designed for a particular group of educators. The Mathematics book is aimed at middle school and secondary school math teachers. The materials given here are from the fourth of the eight two hour inservice sessions.

Preface

The purpose of this Notebook is to assist educators who are designing and implementing inservice education programs to facilitate the effective use of Computer Integrated Instruction (CII) in schools. CII is the use of computer-as-tool to aid problem solving. CII includes use of such generic applications as databases, graphics, spreadsheets, telecommunications, and word processors; these are generic in the sense that they can be used in many different subject areas and grade levels. CII also includes use of special purpose software designed to help solve the problems occurring in specific courses or disciplines.

This Notebook was prepared by the staff of NSF Project TEI 8550588, which received three years of funding beginning 15 September 1985. It is one of four Notebooks to be prepared. The notebooks are:

* CI^3 Notebook for Elementary School
* CI^3 Notebook for Secondary School Mathematics
* CI^3 Notebook for Secondary School Science
* CI^3 Notebook for Secondary School Social Science

The problem addressed by this NSF Research and Development Project is the disparity between the overall capabilities and potentials of CII and the current implementation levels of CII in our schools. There is strong support from computer knowledgeable educational leaders for increased use of CII. Growth in appropriate use of CII depends on schools having:

1. Access to appropriate hardware.
2. Access to appropriate software.
3. Access to appropriate curriculum and instructional support materials.
4. Appropriately trained teachers and school administrators who support increased use of CII.

The cost of computer hardware is continuing to decline even as its capabilities are continuing to increase. The amount of computer hardware available for instructional purposes is now sufficient to begin having a significant impact on schools. Moreover, hardware availability is continuing to grow very rapidly. This project assumes that the problem of access to hardware will gradually diminish; thus, this project does not focus on the hardware problem.

The quantity of educational software is continuing to grow, while the average quality is continuing to improve. A 1986 estimate suggested that there were about 10,000 educationally oriented, commercially available software programs for microcomputers. *The Educational Software Selector*, published by EPIE, lists nearly 8,000 titles. The amount and quality of CII software now available is adequate to support extensive use of CII in schools and to have a major impact on school curricula. The educational market is large enough to support a viable, competitive industry with many companies participating.

This project does not focus on the overall problem of educational software for use in schools. However, each Notebook contains information on a number of pieces of educational software. To the extent possible, the focus is on currently available generic CII software. When more specific pieces of CII software are discussed in the Notebooks, these have been selected because they are readily available, they are apt to remain available for quite a few years, and they fit the specific instructional needs of the authors of these Notebooks.

Instructional support materials include textbooks, workbooks, and reference materials; films, filmstrips and videotapes; and course goals, course outlines, and teacher support materials. While there is a substantial amount of instructional support material for learning/teaching about computers (teaching computer literacy, computer programming, and computer science), the amount of instructional support material for CII is still quite limited. This project includes the development of a modest amount of teacher support material for CII. For example, a number of sample lesson plans have been developed and are included in the Notebooks. However, it is not a primary focus of this project to develop CII instructional support materials.

The primary focus of this NSF Research and Development project is to develop effective methods for the inservice education of educators interested in CII. The materials contained in the Notebooks are designed to be used by computer education leaders who are designing and implementing CII-oriented inservice education workshops and courses. Each Notebook contains a detailed outline of an eight-session workshop, along with supportive materials. For example, there are a number of Copy Me pages that are designed as possible handouts in an inservice workshop or course.

It is recognized that designers and deliverers of inservice education vary widely in experience, computer background, and academic areas of specialization. Two general methods are envisioned for using the materials in these Notebooks. First, an inservice provider might rely very heavily on a particular Notebook, following it quite closely in giving a sequence of workshops or a course. Second, an inservice provider might use these Notebook materials to get ideas and to serve as a resource in designing and presenting CII-oriented instruction to educators. In either case it is expected that the inservice provider will benefit by use of the Notebook materials and will learn some new ideas about effective inservice and CII.

This Notebook presents a particular philosophy of inservice education. It is a process-oriented philosophy, as distinguished from a content-oriented philosophy. Thus, an inservice education program based on this Notebook will look quite a bit different than the traditional computer-oriented inservices that have been widely presented in recent years. The resulting inservices are fun to lead and fun to participate in. They are an effective way to facilitate increased appropriate use of computer-integrated instruction in schools.

5.1

Narrative Overview

This material is from Math Session 4: Problem Solving in the CI^3 Mathematics book developed under NSF project TEI-8550588. The CI^3 Mathematics book is written for inservice presenters. Each session contains a large amount of material to be used during the inservice session and additional materials to be used by participants in their teaching situations.

Problem solving is a buzz phrase in mathematics and in other disciplines. The back to basics movement of the past couple of decades is now being countered with a problem solving movement. In essence, our educational system is struggling to find an appropriate balance among the lower-order skills and the higher-order skills in Bloom's taxonomy. The majority of educational leaders believe we moved too far toward the lower-order skills during the past two decades and that we should now move toward the higher-order skills.

Problem solving tends to have a variety of meanings even in math education. If time permits, it is an interesting exercise to have each inservice participant write down a short definition of what problem solving means to him or her. Then debrief, first in triads and then in the whole group. You may find that few of the participants can give a reasonably comprehensive definition of problem solving! If so, refer them back to Section 2.2 on problem solving and to the four-part math problem solving model discussed in the first of these math inservices.

It is generally agreed that one of the main goals in math education is to increase the math problem solving skills of students. In addition, most math educators agree that students should learn to apply their math problem solving skills to math-oriented problems in other disciplines. But this turns our to be a major difficulty. For example, many students will study metric measurements in a math class and demonstrate competence through their performance on a test. They will then go into a science class and display complete ignorance of metrics!

Learning theory talks about *transfer* of learning. The transfer of metric knowledge from a math class to a science class is difficult for many students. But research on learning theory suggests that transfer can be increased by appropriate *teaching for transfer*. One main thrust of this inservice session is the question of transfer and how to help increase it. One good way to start this sesion is to briefly discuss the topic and then ask participants to write down some examples of what they do to facilitate transfer. Debrief in triads and in the whole group.

There are many pieces of problem solving software available. Participants will get a chance to use several during this inservice session. Generally the problem solving software has been designed in a game format and is quite a bit of fun to use. Fundamental questions are:

1. What does a student learn about problem solving by using the software?

2. To what extent does this knowledge transfer to:
   a. Solving math problems in a math environment?
   b. Using math to help solve problems in non math areas?

The first piece of problem solving software examined in this inservice is quite simplistic. But it suffices to allow discussion of the above questions and then to move into other, related questions. For example, suppose that a studnet develops good skill at solving computer based maze problems. Does this help improve spacial visualization skills or planning-ahead skills? (Pat McClurg did her doctorate work at the University of Oregon on a similar question. She found that when students used software such as *The Factory* over a period of weeks, their scores on spacial visualization tests went up more than did those in a control group.)

In this inservice session (as in all sessions) we stress participants working together at the computers. There has been quite a bit of research on cooperative learning and cooperative problem solving. The research supports the contention that students will learn more about problem solving by working together to solve problems. This might be raised as a discussion topic in one of the debriefings!

The December/January, 1986-87 issue of *The Computing Teacher* contains an article titled "The Teacher's Role in Using the Computer to Teach Thinking Skills." That article is in the Readings section of this inservice session.

Some Specifics on this Session

The session begins with an off-machine problem solving activity. There are many books full of such activities. A few sample activities have been included in this part of the Notebook. Select one that you feel will be appropriate to the participants in the inservice.

Spend a short time with the activity and then debrief the steps the participants took in order to solve the problem. Cooperation and working together are needed in some, while other activities can be done alone. Raise the question of whether the problem solving ideas they used carry over to solving regular math problems.

The second activity is done at the computer. Have the software *Mind Puzzles* by MECC already loaded so that participants can begin at once. Tell the participants that they will have approximately 20 minutes to get through the software. They should do both *Maze of Rodentia* and *Queen Bee of Menta. Maze of Rodentia* is very straightforward and the directions are quite explicit. It is a maze game and should be easily understood. *Queen Bee of Menta* is not as easily understood and may take some help with individuals. Once the objective is understood, the game itself is also very easy.

Mind Puzzles was picked because it's easy to use and it is a tool to help raise questions concerning problem solving. The debriefing for this section can take more time than the time spent on the computer. We can use this piece of software to raise questions such as:

1. Is it appropriate to use games in teaching?

2. Is this a good piece of software? (How does one judge whether a particular piece of problem solving software is good?)

3. Does everyone (teachers included) always read directions? Is it possible to write directions that all readers will find to be clear?

The third activity is done after the break. A number of computer stations are loaded with a variety of problem solving software (Do this during the break. Get some volunteers to help and to serve as assistants at the stations.) There should be software for the low- as well as the high -ability students. Participants will get a chance to visit several stations.

Be sure to allow some time for debriefing, even if this means that participants don't get to visit every station. (Participants can explore software on their own, but it is harder for participants to have group discussions on their own.) Make sure that transfer of learning gets emphasized in the closing debriefing.

5.2
Script

Theme: Using problem solving software.

Objectives:

1) To have participants think about what is meant by problem solving and uses of computers to create environments that support the development of good problem solving skills.

2) To increase knowledge of the issue of transfer of learning and to focus on how to increase the amount of transfer that occurs as students use problem solving software.

3) To explore some problem solving software.

Materials:

Software	Handouts
Mind Puzzles	Colors of a Cube
Teasers by Tobbs	Off-Line P.S. Activity with jigsaw pieces
The Factory	Data record sheets
Gertrude's Puzzles	Description of each piece of software demonstrated.
High Wire Logic	Article from *The Computing Teacher*
Rocky's Boots	

Pre-Class: Load *Mind Puzzles* into the computers. The data record sheets should be included with participant handouts.

Introduction:
5-10 minutes

Start with the Colors of a Cube worksheet and let the participants work on it. Do not give directions. Let participants work alone or in groups. Once an individual or group solves the problem, watch to see if they help each other, give the answer away, or just sit quietly. Use your observations in the debriefing.

Discussion:
10-15 minutes

Debrief the above activity. How did the group react to the problem? What are we trying to accomplish when we do problem solving activities? Was the activity motivating? How do you think students feel when they are faced with these types of problems? Do these types of activities help students use what they have learned in a math class? Have participants keep these questions in mind when they go to the next activity.

Activity:
20-25 minutes

Purpose: Have participants use a problem solving program and think about what steps they follow. Have participants work in pairs at the computer. Tell them that they will have approximately 25 minutes to go through both programs. Give no instructions verbally; participants should have the instruction sheet with them from the handouts.

Discussion:
15 minutes

Debrief the above activity. How was this different from the worksheet they received at the beginning? What skills were being used? Was the computer useful for this activity? Can this be useful for students in other areas or is this "just a game"? Did they learn anything from this piece of software? Was it fun? Would they use it anywhere in their curriculum or elsewhere? Try to get everyone involved in the discussion. There are always a few people who have definite opinions, but we are trying to facilitate new thinking and want to get everyone consciously thinking.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.2 Script
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Break: 10 minutes

Collect logs.

Activity: 30 minutes

Purpose: To allow the participants to see other types of problem solving software and have some hands-on time. It is desirable to have some of the inservice participants serve as facilitators at the stations. Have at least four different pieces of software available for teachers to try out. Have participants pick and choose which ones they would like to see, but keep the group from gathering all at one computer.

Discussion: 10 minutes

Debrief the above activity. Tell them that there is a list of problem solving software in the handouts. They can mark the ones that are available to them (check ahead of time at resource centers, ESD's, etc., so that you can let them know). The list given here is not intended to be comprehensive, and new pieces of this type of software are being published quite regularly. Ask the participants if any of them have used a piece of problem solving software that they've found to be particularly appropriate to their teaching situation.

Close the session by briefly reviewing what was discussed. Problem solving and transference should be something teachers are conscious of and work toward directing students to do on their own.

Set the time and place for the next session.

5.3
Timeline

This is a sample timeline for a two-hour inservice based on materials in 4.4.2. The main presenter was Tsukiko Harriman; she was assisted by Sue Hickey. Have *Mind Puzzles* by MECC ready and loaded into the computers. A large color screen or monitor is needed for this session.

0:00 - 0:08	The participants will enter and begin work on the Colors of a Cube activity. They may work alone or in groups. Do not give any specific instructions.
0:08 - 0:15	Debrief the above activity. What do we hope to accomplish by having students do these types of worksheets? Does the learning transfer to other math problem solving situations, either in math classes or in other problem solving situations? (Tsuki)
0:15 - 0:40	Activity 1. Have the participants pair up and go to computers. Have *Mind Puzzles* already loaded and ready to go. Tell participants they will have 25 minutes to go through both games. They should make notes of problems, questions, answers, etc., as they work with the software.
0:40 - 0:50	Debrief the above activity. How many people read directions first as opposed to trying to do the work without reading them? How was this activity different from the worksheet activity used at the beginning of the session? What skills were being used in solving the problem? What did they learn from doing this activity? Was it fun? Will the learning transfer? What could teachers

do to increase the transfer? (Tsuki)

0:50 - 1:00	**Break.** Collect logs and Game of the Week sheets.
1:00 - 1:10	Debrief Game of the Week. (Tsuki)
1:10 - 1:20	Demonstration using *Rocky's Boots*. Model how one might use a single computer in one's classroom. Involve the whole group. (Sue)
1:20 - 1:50	Debrief briefly and explain that for the next activity there will be six different stations available, each with one piece of software: *Rocky's Boots, Getrude's Puzzles, High Wire Logic, Teaser's by Tobbs* , *The Factory* and *MECC Math Vol. 1*. They will have 25 minutes to to visit two to three different stations. (Tsuki)
1:50 - 2:00	Debrief the above activity. Which software seemed most useful for working with students? What could students learn from using pieces of software such as these? Will transfer of learning occur, and what can teachers do to increase transfer? Is it okay to play such games and have fun during a math class? Research says that one gets better at problem solving by being in a problem solving environment in which one practices problem solving. Is the typical secondary school math classroom a good problem solving environment in which students can practice problem solving? Research also says that a teacher's attitude toward problem solving and transfer of learning is an important factor. (Tsuki, Sue)

5.4
Handouts

The pages of this section are handout materials needed by the participants during Session 4 of the Math Inservice. The facilitator may find it useful to make some of these into overhead projector foils for use during the inservice.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Colors of a Cube (A Problem/Puzzle)

A cube has six sides, each a different color.

The red side is opposite the black.

The green side is between the red and black.

The blue side is adjacent to the white.

The brown side is adjacent to the blue.

The red side is face down.

What color is the side that is opposite the brown side?

Now that you have finished, answer the following questions.

1. What grade levels of students might be able to solve this problem, and what do you think students would learn from doing this exercise?

2. What skills are used in solving this problem? Name some places in the math curriculum or in applying math to other disciplines where these skills are useful.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Introduction to Mind Puzzles by MECC

Mind Puzzles is a package which helps students develop a variety of problem solving skills that apply to virtually all curriculum areas. *Mind Puzzles* is process oriented rather than subject oriented and can be used wherever the objective is to develop thinking skills and processes rather than to acquire knowledge. The programs are designed for individual or small group use by students in grades 6 - 9. The emphasis is on a discovery approach to problem solving, encouraging students to use the computer as an exploratory tool.

Mazes of Rodentia presents students with a number of mazes of varying complexity. Students must successfully navigate the mazes in order to win a tournament. There are three different winning catagories. To succeed, students must exit a maze with the fewest moves.

In *Queen Bee of Menta* students explore the program to discover the secret code which involves a four by four matrix of symbols and their associated screen displays. Screen displays include characteristic borders, movements, formations and sounds. The exhibition of each type of characteristic is determined by the order in which the symbols are selected.

A *Top 10 Scores* frame is part of each program. A *Teacher Option* enables teachers to view or delete the names of the 10 top-scoring players in each category.

Students can exit the program at any time by pressing Escape twice in a row when the program is waiting for input.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Mind Puzzles (Performance Aid)

I. How to get started:

1. Insert the *Mind Puzzles* disk, close the drive door and turn on the computer.

2. Type in 1 and then press the Return key to choose *Mazes of Rodentia.*

3. Type in Y to the next question.
   Take any notes you may need here:

4. First choose Maze Level 1 (Later choose 2, 3, or 4).
5. Traverse and Record using the next page.
6. What happens when a different aid is chosen?

   How is the highest score obtained?

II. Now to advance:

1. After returning to the menu, choose 2 and press Return for the *Queen Bee of Menta.*

2. Y (Return) to read about the history.
   N (Return) to receive training.

3. Begin as an Apprentice Bee Keeper, Level 1.
   Go through the tutorial.
   N (Return) to the next level of training.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

4. Now become trained as an Apprentice Bee Keeper, Level 2. Use the recorder sheet (page 1a of MECC materials).
5. After that training, accept the challenge of 3 (Return). Use the recorder sheets to gather data about these four new symbols and each of their attributes. Then take the challenge (page 1b of MECC materials).

III. Look through the booklet from MECC for a more thorough understanding of this disk.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Some Problem Solving Software for Math Teachers

Here is a short list of some problem solving software on the market today (1987) that is generally acknowledged to be of good quality.

Sunburst Communications
39 Washington Ave.
Pleasantville, NY 14850

1. The Factory	Spatial relationships, multiple solutions.
2. Geometric Supposers	Geometric concepts.
3. High Wire Logic	Deductive logic, boolean operations, critical attributes.
4. The King's Rule	Inductive reasoning, numerical patterns and relationships.
5. Puzzle Tanks	Math and logic skills, multiple solutions.
6. Safari Search	Deductive reasoning, analysis.
7. Teasers by Tobbs	Problem solving.
8. Semcalc	Working with a variety of units in word problems.
9. Interpreting Graphs	Graphs are given and the user must find the situation that has been graphed.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

The Learning Company
4370 Alpine Rd.
Portola Valley, CA 94025

1. Gertrude's Puzzles	Spatial relationships, critical attributes.
2. Moptown Hotel	Sequences, deductive reasoning.
3. Rocky's Boots	Logic, spatial relationships.

MECC Software
3490 Lexington Ave. North
St. Paul, MN 55126

1. Problem Solving Strategies	Looks at exhaustive search, trial and error, making lists and tables, and simplifying the problem.
2. Math Vol. 1	Eight programs to solve math problems for sixth grade on up.
3. Math Vol. 4	Thirteen programs for Algebra II - Trig students on up.

Conduit
The University of Iowa
Oakdale Campus
Iowa City, IA 52242

1. Graphing Equations (Green Globs)	Problem solving through graphing.

Rocky's Boots

(Grade 2 - College)

Rocky's Boots is a piece of problem solving and logic forming software. It is in a game format where the player moves from room to room as he or she learns what each piece does and how to connect them. There are six sections, each with various numbers of rooms, increasing in the level of difficulty as you go up. Some of the skills involved are: activating targets, identifying positive and negative values, recognizing and avoiding glitches, building and debugging circuits, classifying, abstracting and inferring, applying Boolean logic to solve problems, and using combinatorial logic.

When using this piece of software for demonstration, it may be helpful to have pages 25 - 26 of the manual that accompanies the software copied for a handout. These pages give a screen picture and a short description of each section. You will need a color monitor to use this piece of software.

The package comes with a disk and a student handbook. There are versions available for Apple, Commodore, IBM, and Tandy computers. *Rocky's Boots* retails at $49.95 and is available from:

The Learning Company
545 Middlefield Rd
Suite 170
Menlo Park, CA 94025
800/852 - 2255

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Gertrude's Puzzles

(Grades 1 - 6)

Gertrude's Puzzles is a problem solving program. It builds basic thinking skills in a game format. It is an exploratory game in which students solve puzzles. To play the game and solve the puzzles, players must understand and use the attributes of the playing pieces on the screen. They do this by following rules of classification and ordering based on color and shape. As they move through the various rooms, they can control the environment by choosing which puzzle to solve and its level of difficulty.

When using this piece of software for demonstration, it is helpful to have pages T7 - T9 from the documentation ready for handouts. These pages contain screen pictures of the eight rooms as well as a short explanation of what can be done in each of the rooms. You will need a color monitor to use this piece of software.

Gertrude's Puzzles comes with a disk and student handbook. There are versions available for Apple, Commodore, and IBM computers. *Gertrude's Puzzles* retails at $44.95 and is available from:

The Learning Company
545 Middlefield Rd.
Suite 170
Menlo Park, CA 94025
800/852 - 2255

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

High Wire Logic

(Grades 5 - 9)

High Wire Logic is a game based on Boolean logic. There are two options in a game setting. Some of the skills involved in this piece of software are working with higher order rules, identifying multiple solutions, scanning for clues and hints, making organized lists, and examining assumptions.

The Practice option introduces and allows students to practice six logic types: and, or, and-or, and-and, or-or, and the exclusive or (xor). The Play option shows students two sets of colored shapes; one set is up on a high wire, the other set falls to the net below. The shapes on the wire fit the rule. The shapes in the net do not fit the rule. Students are challenged to write as many rules as they can that fit the shapes on the wire but not the shapes in the net. The more difficult the logic type used, the higher the points awarded for each correct rule.

When using this piece of software for demonstration, it may be helpful to have pages 4 - 8 from the documentation ready to pass out to participants. These pages have screen pictures and a short explanation of each rule. You will need a color monitor to use this piece of software.

High Wire Logic comes with two disks and a teaching guide. It is available for the Apple computer and retails at $59. It is available from:

Sunburst Communications
Room D 7575
39 Washington Ave.
Pleasantville, NY 10570
800/431 - 1934

Teasers by Tobbs

(Grades 4 - 10)

Teasers by Tobbs is a piece of problem solving software that provides drill and practice for addition and multiplication facts. The game is in the shape of an addition or multiplication grid. The grid presents a series of math problems, each with a missing number. Students decide which number can't be, might be, or must be the correct solution. Students can play solo or in groups, and a score is kept. There are six levels, each one advancing in difficulty.

Teasers by Tobbs comes with two programs, a disk, a teaching guide, and a back-up disk. There are versions available for Apple, Atari, TRS-80, IBM, and Commodore computers. *Teasers by Tobbs* retails at $59 and is available from:

Sunburst Communications
Room D 7575
39 Washington Ave.
Pleasantville, NY 10570
800/431 - 1934

The Factory

(Grades 4 - 12)

The Factory is a problem solving visualization game. It is designed to teach several strategies: working backward, analyzing a process, determining a sequence, applying creativity, visual reasoning, and rotation. There are three levels. The first level allows students to experiment with three machines. The second level allows a student to make a product and challenge other students to produce it. The third level has the computer produce a product and then the student must produce it.

The Factory comes with three programs, a disk, a teaching guide, and a back-up disk. There are versions for Apple, Atari, Commodore, TRS-80, and IBM computers, and there is a network version available. The Factory retails at $59 and is available from:

Sunburst Communications
Room D 7575
39 Washington Ave.
Pleasantville, NY 10570
800/431 - 1934

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

MECC Math Volume 1

(Grades 3 - 6)

This software package gives students experience in using basic logic to master elementary math concepts. There are three different games available. Students interpret clues as they try to guess a secret number, track down a *hurkle* hiding on a coordinate grid, or confront a factor-grabbing fiend.

MECC Math Vol. 1 comes with a diskette and a teacher's guide. There is only a version for the Apple computer. *MECC Math Vol. 1* is available from:

Minnesota Educational Computing Consortium
3490 Lexington Ave. North
St. Paul, MN 55126
612/481 - 3500

Note that many states, provinces and school districts have MECC site license contracts. The license allows multiple copies of software to be made for use in classrooms.

Your Name ____________________

Game of the Week

1. Ask your students to give you an example of a real life situation outside of school in which they have had to use the types of math skills and content being taught in the math course they are currently taking. Also, ask them to provide an example of an application of these math skills and knowledge that they have made in some other class (not the math class) sometime during the current school year.

OR:

2. Write down some examples of what **you** do in the classroom that facilitates transfer of learning. (If you have never asked your students what could be done in class to help increase transfer of learning, you might try doing that and then discussing the results.)

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Game of the Week Outcomes

Note: This is not intended as a handout. Rather, it is a brief discussion to give the inservice facilitator some ideas on what outcomes may be expected from the Game of the Week. There were two choices for this week's game.

1. Ask your students to give you an example of a real life situation outside of school in which they have had to use the types of math skills and content being taught in the math course they are currently taking. Also, ask them to provide an example of an application of these math skills and knowledge that they have made in some other class (not the math class) sometime during the current school year.

To a large extent, students give responses bases on math skills learned prior to their current course. Typical answers from middle school students (grades 6-8):

- +, -, x, / of whole numbers, decimals and fractions
  - shopping
  - counting items
  - keeping score
  - checkbook balancing
  - saving money
  - differences in dates/time
  - home ec/cooking/calories
  - dividing people into groups
  - bills
  - how many pages left to read
  - locker combinations
  - telling time
  - checking lunch money
  - taxes
  - profit making
  - comparative shopping
  - science - timing/measuring
  - figuring out grades
  - averages/sports/pop. density
  - interest
  - estimating
  - mileage/distance
- teaching
- computer class/games
- playing chess/cribbage/monopoly
- measuring/metal shop/building something
- finding area/scaling/drawing/architecture
- computer circuits on a chip

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

High school students have answered as follows:

- finding discounts
- centering a mirror/carpentry/architecture
- probability questions
- programming graphics on a computer
- analyzing problems for solutions/what method to use
- reading and interpreting tables
- formulas and when to use them
- doing two-column and algebraic proofs
- making graphs/scale drawings/maps
- in science/forensics
- planning a trip/distances
- proportions
- changing Farenheit to Celcius
- cooking
- paychecks
- surveying
- rate of change
- area/volume
- games/sports
- enough money/age/bank accounts

2. Write down some examples of what **you** do in the classroom that facilitates transfer of learning.

- recipe adjustment
- kids compute own percentages for quiz grades
- give practical, real life situation story problems and problem solving problems
- simulations (i.e., cost of a home with interest held constant)
- have them look for patterns, guess at answers, sketch
- break down a difficult problem into a few, easier problems
- give questions resulting from the answer to the problem to make them think about the problem
- numerical guessing games

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.4 Handouts
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

- cut and paste/making models with polygons
- use materials from the real world (newspapers, checks...)
- do scientific experiments that relate to math formulas or concepts
- application to real life situations dealing with students whenever possible
- make assignments where students must use prior knowledge and the current lesson to complete it.
- average and sort data

5.5
Materials

These materials are samples of lesson plans designed to be used with students at a variety of grade levels. They represent examples of the types of activities that can be done with students after inservice participants have learned the material of Session 4.

Inservice facilitators may want to make use of some of these materials during the inservice session. If the session is more than two hours in length, it is appropriate to have participants work through a number of these student activities.

Off-Line Problem Solving Activity (Jigsaw Puzzle)

Description:

Topic: Problem solving - Hypothesis Posing
Grade level: 2 - 12
Time: 3 - 5 minutes
Grouping: Whole group
Materials: Letters that spell a word with each letter cut into puzzle pieces.
Objectives: 1) No single student can solve the problem from the start.
2) All students participate.
3) Students develop group-building skills of communication, cooperation and trust.

Before you start:

Cut out the letters of the word selected. Cut each letter using a jigsaw-type cut.

Lesson:

1. As students come into the classroom, hand each student one piece. Do NOT explain the problem to them.

2. Make observations about what students are doing with pieces and state those out loud. "Jane is holding up her piece." "Jack is turning his around and flipping his piece."

3. Encourage anyone who is trying to combine two pieces. Allow students to move to combine pieces or check a fit.

4. Once a couple of students begin to recognize that letters can be formed, encourage the rest to engage in forming letters.

5. Usually at this point students decide that the letters can be combined to spell a word. If not, a question to ask might be "What do you do now with these letters?" Typically, someone takes over leadership and arranges the letters. The teacher can then write the letters on the

board for all in the class to see.

6. Debrief the activity. Have students verbalize the steps in the problem solving process. Reinforce the concept of problem posing. There were two problems: what the pieces were for and what the letters combined to make. Reinforce the groups cooperation, communication, and trust.

Followup: (Other problem solving activities)

1. Ask students to arrange themselves alphabetically without speaking.

2. Blindfold students and ask them to arrange themselves by height without speaking.

3. Form a circle of four students holding hands. Bring all hands to the center, let go of those hands, and take hands from elsewhere in the circle (no two people holding only each other's hands). Now untie the knot to form a circle without letting go of hands. Some students may end up facing the outside of the circle. Now form a group of five, six, or more students and repeat the activity. Is it possible to untie a circle of 12 students?

4. Any problem which meets these conditions:
   a) No one person can solve the problem completely by himself or herself.
   b) Communication is required in some form for the problem to be solved.
   c) Movement of materials or students is required.
   d) Takes less than five minutes.

5. For additional problems, have students design and then execute them in the classroom.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.5 Materials
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

MECC Math Vol. 4

Description:

Topic: Problem Solving
Grade Level: 11 - College
Time: Varies
Grouping: Demo or Individual Use
Objectives: Using the computer to solve complex equations.

Materials:

Software: *MECC Math Vol. 4*

Equipment: Apple and Monitor. Data Disk, Printer (optional)

Before you start:

Select problems that would be very time consuming to solve manually.

Lesson: (Sequence of instructional activities)

-Students then use the proper program to solve the problem.
-Problems should be posed which encourage changing the parameters or function values.
-Use real world problems and have students find methods for solving the problem. Have them compare the efficiency of the various methods.

Followup:

Have students find real world problems that would be very difficult to solve without the help of these programs.
Where are other areas of mathematics that should use the computer as an aid to solve the problem?
Where in the real world is the computer used to solve problems?

Comments:

Each of these programs has very specific formats. It will be necessary to spend a little time to understand the proper inputs.

The Factory

Description:

Topic: Thinking Backward
Grade Level: 3-12
Time: One or more 15 min sessions
Grouping: Whole class or groups of 2 or 3
Objectives: 1) Thinking Backward Strategy
2) Exercise Spatial Relations
3) Sequencing Steps Strategy

Materials:

Software: *The Factory* by Sunburst

Equipment: One computer or many

Other materials: Concrete aids or problems

Before you start:

1. Have students use the first two parts of the menu to learn how each machine operates and how to design a product.

2. Do a problem in class that is best done by thinking backward. Here are two examples:
   a) Seymour left a plate of brownies to be equally divided between Jim, Judi and Larry. Jim took his share to his room. Judi did not know Jim had any so she took what she thought was her share. Larry came in and, not knowing that Jim and Judi had taken cookies, took his share. This left eight cookies. (Answer not given; this could be done with real brownies.)

   b) Begin with a graph and figure out the equation that the graph represents.

3. Do a problem that requires some spatial manipulating. Examples may be found in many different puzzle books.

4. Using *The Factory*, under Make a Product, have the whole class determine how to make one easy product. This allows you to work on their vocabulary, discuss sequencing, and verbalize strategies.

5. Have the students close their eyes. You describe three machines in a particular order. Students are to visualize the product that the machines would produce and then draw their predictions on paper. Then use *The Factory* to check their predictions.

Lesson: (Sequence of instructional activities)

1. Working in groups of two or three students, go to the machine and design an easy product.

2. After all have designed an easy product, have groups of two or three create a medium or difficult products.

3. Have students design a factory on paper. Predict their product. Test out the factory design and check how close they were to the prediction.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.5 Materials
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

4. Have students challenge others to design a factory that produces their product.

5. Have students come up with other examples in everyday life that require the Thinking Backward strategy.

Followup:

1. Select problems that would be very difficult to solve without lots of calculation or time and have students identify when Thinking Backward is the appropriate strategy.

2. Do some imagery exercises with various problems in the future. Have students close their eyes and form the appropriate images for a certain setting and then begin the lesson.

Comments:

Visualizations are helpful in developing concepts. They provide a deeper understanding of mathematics. The ability to visualize and manipulate shapes can be developed through practice.

Problem Solving Strategies: Diagonals

Description:

Topic: Problem Solving Strategies
Grade Levels: 5 - 9
Time:
Grouping: Demo; divide the students into groups of four.
Objectives: Learn three problem solving strategies:
1) Trial and Error
2) Simplifying
3) Exhaustive Listing

Materials:

Software: *MECC Problem Solving Diagonals*

Equipment: Apple Computer, Monitor

Other materials: Dodecagons drawn on handouts for student

Before you start: This can be part of a problem solving unit or done as an individual lesson. You can review basic geometric parts -- sides, vertices, and diagonals -- before the lesson or let the program do it.

Lesson: (Sequence of instructional activities)

Demonstration: Use the tutorial aspects of the program and use the program as a demonstration. The students will work on the problem individually as the program asks questions. The sequence of steps is:

1) Have you used this program before? Answer No. Then do the tutorial on dodecagons.
2) Program will go through the following strategies:
   a) Trial and Error - Have the class guess the number of diagonals.
   b) Make an Exhaustive Listing - Students work at their desks and try to list all the diagonals. Do they get them all? Leave some out?
   c) Simplifying the Problem - Students will make a table and find the number of diagonals in a triangle, square, pentagon, hexagon, etc., trying to find a pattern.

Student Groups: Divide the class into groups of three or four students.

1) Teacher introduces the problem, reviews the vocabulary (dodecagon, sides, vertices, diagonals).

2) From the main menu of the program *Diagonals,* choose the strategies you wish to explore.
   a) Trial and Error - Have each group guess the number of diagonals. Ask if they used any strategy in their guessing.
   b) Exhaustive Search - Divide the groups so that there is a record keeper to keep track of all of the diagonals.
   c) Simplify the Problem - A record keeper is needed to make a table.

Page 7
This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.5 Materials
NSF Project TEI 8550588: Computer-Integrated Instruction Insr ...se

Everyone in the group attempts to find the formula for the pattern.

3) Ask class to decide when the different strategies are most appropriate.

Followup: Have the class extend their answers to find the number of diagonals in 15-or 20-sided figures. Use these strategies with other kinds of problems.

Comments: This program could be run by groups of students as a tutorial. This well-directed program does not need teacher supervision.

Teasers by Tobbs

Description:

Topic: Problem Solving
Grade Level: 3 - 9
Time: 5-7 min./level
Grouping: Two per computer
Objectives: Use problem solving strategies such as
1) Guess and Check;
2) Inductive Reasoning;
3) Thinking Backwards

Materials:

Software: *Teasers by Tobbs*

Equipment: Apple Computer

Other materials: scratch paper

Before you start:

Students should know their addition facts before using the addition portions and their multiplication facts before using the multiplication portions.

Teasers by Tobbs is sequenced to take the user from beginning drill through deductive reasoning. The reinforcing drill gives students reinforcement of their arithmetic facts and time to adjust to the format of the program and the computer. The later levels of difficulty present problems in a backwards presentation and lead into deductive activities.

Lesson: (Sequence of instructional activities)

1. Students should proceed through the different levels at their own pace.

2. The teacher can use the program for demonstration, reinforcement and strategy clarification at different points.

3. Instead of asking for the answer, ask for the two numbers that form the answer where Tobbs is sitting.

4. This can be used in an introductory Algebra course, as the higher levels introduce solving equations and the concept of variables.

This material is on the Macintosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.5 Materials
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Followup:

Have students design arithmetic grids similar to those in the program to challenge others in the class or other classes.

Comments:

Stress the analysis of the strategies used to solve the problems. Get students to verbalize the strategies and understand that multiple correct solutions exist to any given problem.

This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.5 Materials
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

Tic Tac Show Lesson Plan

Description:

Topic: History, Science, or anything else
Grade Level: 4-5
Time: 20 min + 20 min + 20 min
Grouping: Two or three students working together
Objectives: Students design a quiz, take the quiz, and revise the quiz

Materials:

Software: *Tic Tac Show* by Advanced Ideas

Equipment: One or many computers

Other materials: student questions on paper

Before you start:

This can be one group or many different groups.
Students will design a quiz containing questions relative to a specifc topic and supply all the right answers to the questions. They must keep in mind the questions will be given to the whole class.
Select the topic or source of content to be quizzed.
Students should have played Tic Tac Show.

Lesson: (Sequence of instructional activities)

1. Students write 10 questions.
2. They write all the possible correct answers to the 10 questions.
3. They enter the questions in the editor fc. *Tic Tac Show* and then save the file on disk.
4. They play the game and check their quiz with correct and incorrect answers, taking notes on any possible changes.
5. Students then go back to the editor and revise their questions and answers.
6. The teacher now plays *Tic Tac Show* with the whole class as a review before a quiz.

Followup:

1. Quiz someone else's room with the material.
2. Have students from another room design a quiz for your room.
3. Have the students present the quiz at a PTA or PTO meeting.

Comments:

The major purpose of this lesson is to have students analyze their wording and become comfortable with revision. Some additional products include the realization that often questions have more than one acceptable right answer (e.g. What is 3+4? Seven or 7 are both correct), analysis of what content is important in a section and student cooperation.

Page 11
This material is on the MacIntosh disk.
Folder 4 Math Subfolder 4.4 Problem Solving File 4.4.5 Materials
NSF Project TEI 8550588: Computer-Integrated Instruction Inservice

5.6

Readings (Number 1)

The Teacher's Role in Using the Computer to Teach Thinking Skills

by

Robert M. van Deusen and Jean Donham

(From the December/January 1986-87 issue of *The Computing Teacher*, reprinted with permission from ICCE.)

We teachers have complained about the quality of instructional software for some time. But our complaint is becoming less valid as software developers have begun to identify a curricular niche for the computer: thinking skills. Attempts to teach thinking skills in the context of social studies, science or literature are sometimes unsuccessful because in the struggle between teaching *content* and teaching *process*, content often wins. What is needed is a manageable environment in which the focus can be placed on process -- the thinking skills.

How Do We Create Such an Environment With a Computer?

As educators, many of us have tended to assume that for students to learn with a computer, they must be seated at it, hands on. There are indeed situations in which that is appropriate. However, if the teacher is to play an active role in focusing on skills and strategies, then the teacher must be interacting with students and the computer. To teach thinking skills, such teacher direction is essential. A 25-inch monitor for the students, one computer, and a small monitor for the teacher is an appropriate arrangement. The teacher can then run the program and direct the instruction, focusing on the targeted thinking skills, labeling those skills, and increasing the probability of transfer.

Appropriate software can help the teacher create a manageable environment to focus on thinking. The following criteria can assist in choosing this software:

- Does the software focus on significant thinking skills as its objective?
- Are there multiple levels of difficulty so that the teacher can adjust to students' needs?
- Are there multiple solutions which vary in sophistication?
- Does the software have multiple problems so that it can be used repeatedly?
- Is the software designed for a cooperative, noncompetitive environment?

Most software designed to teach thinking skills does so within a specific situation that can be effectively modeled with the computer. Figure 1 lists examples of exemplary software with which to teach thinking skills. Avoid the assumption that software teaches. It is the teacher who teaches; the software is only the medium.

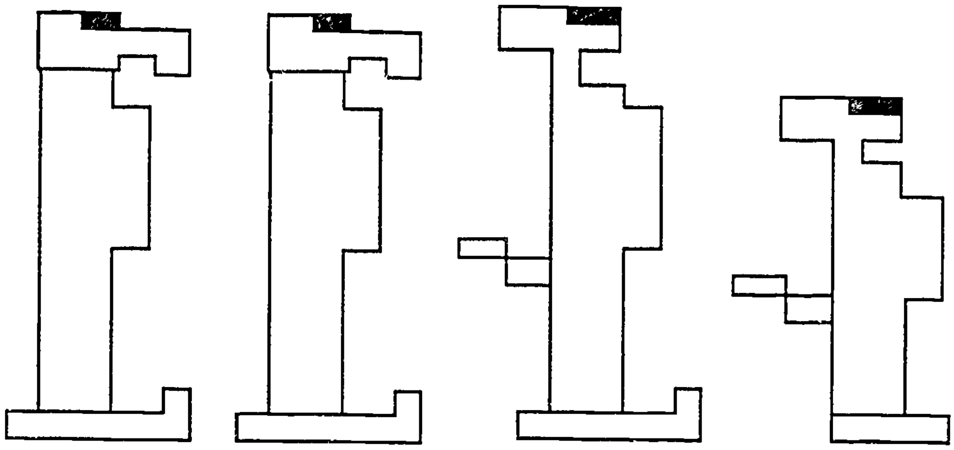
What Does the Teacher Need to Do?

As with any instructional material, the teacher must personally learn how to use the software first. Learning the software is always composed of two parts: learning the mechanics of the levels, choices, exit mechanisms, and help systems which allow one to move between the various components of the program; and understanding the learnin objectives of the program -- What is it designed to teach? How can the teacher facilitate that learning? And what kind of questions should the teacher ask which will cause students to actively develop the structures and insights this program is designed to encourage? This second component -- the teaching component -- is often totally neglected. This often occurs when teachers assume that software teaches. Teachers teach, and in order to effectively do so, they need to know where they are going or they're liable to end up somewhere else.

"Where teachers are going" is in search of thinking processes, not in search of right answers. Consider, for example, *Safari Search* from Sunburst Communications. *Safari Search* was designed to teach deductive logic, analysis, and hypothesis formation and testing. A five-by-five grid is presented, behind which one or two animals hide. Each cell of the grid will display, upon being chosen, relevant information about the direction or proximity to the hidden animal(s). It is, of course, the *process* of determining the location of the animals, the language used to explicate that process, and the thinking behind that language, which the teacher wants students to experience.

If the teacher has worked through the program as a *learner*, and then stepped back to analyze the process used, it is often clear that a model or manipulative would aid the student while using the software. Physical models or paper/pencil, two-dimensional, semi-concrete models can aid in the development of the concept or strategies being learned.

For example, when using The Learning Company's *Moptown Parade* at Level 6, it is very desirable to use physical manipulatives of the 16 possible Mopppets (two heights, two weights, two colors, and two kinds). In Level 6 students line up the Moppets, changing one, two, three or four attributes between each of the five Moppets in the parade. Learners benefit significantly by actually moving the manipulatives and checking to see if each differs by the specified number of attributes. While some students will limit themselves to a guess and check strategy, others will progress to analysis, generalization, elaboration, and synthesis through the use of manipulatives. This focus on the *process* of learning through actively *manipulating the environment* and *creating* their own knowledge allows students to connect the artificially divided compartments of learning into a fabric of related elements.



Tall Fat Red Bibbit | Tall Fat Blue Bibbit | Tall Fat Blue Grippit | Short Fat Blue Grippit

While working through the learning objectives, the teacher may divide the students into cooperative groups to gather and organize data, form and test hypotheses, make generalizations, elaborate, find multiple solutions, or even write an algorithm for the general solution to the class of problem. Cooperative groups allow students to develop their own theories on the problem and its possible solutions. In addition, students who are frustrated at a given stage in the problem solving process have the opportunity to retrace and clarify the process with peers. Cooperative groups also limit the impact of those students who might divulge the essence of the solution before others have had time to process.

Program	Targeted skills
The Factory (Sunburst Communications)	Spatial relationships, problem solving, multiple solutions
The Incredible Laboratory (Sunburst Communications)	Hypothesis testing and formation, organization of data
Safari Search (Sunburst Communications)	Deductive reasoning, analysis
The King's Rule (Sunburst Communications)	Inductive reasoning, numerical patterns and relationships
Moptown Parade (The Learning Company)	Sequences, problem solving, deduction, critical attributes
Teasers by Tobbs (Sunburst Communications)	Problem solving
Gertrude's Secrets (The Learning Company)	Spatial relationships, critical attributes
High Wire Logic (Sunburst Communications)	Deductive logic, Boolean operations, critical attributes
Word-a-Mation (Sunburst Communications)	Word relationships, pre-analogy
Puzzle Tanks (Sunburst Communications)	Problem solving multiple solutions

Figure 1.
Software for Teaching Thinking Skills

The teacher's role in questioning throughout the process is critical. Thinking out loud is an important aspect of teaching thinking skills. Teachers can demonstrate effective strategies for problem solving by creating an environment in which all thinking is thinking aloud. Often a three-step questioning approach is useful. "*What* do you think?" establishes the students' positions. Whether right or wrong, the follow-up needs to be asked, "Why do you think it?" This elicits the rationale behind the students' thinking. Follow-up questions which explore exceptions, special cases and apparent contradictions will cause students to push their own thinking as far as it can go. The third question, "How did you figure it out?" asks students to relate the steps or processes used in arriving at the stated position.

The teacher must pursue all three steps. Whether the initial answer to the "What do you think?" question is *right* or *wrong* is irrelevant. The answers to the final two questions are much more revealing than the answer to the first question.

The teacher's role in the model shifts from evaluator to facilitator. A metaphor may help to clarify the distinction. The leader of a mountain-climbing expedition has choices: He or she can climb independently to the top and extol the beauty and virtues of the mountain, and then try to pull everyone else up the mountain; or he or she can work with the climbers every step of the way as a fellow mountain climber, albeit one with more experience and knowledge. It is this latter technique that the computer allows, with teacher and students as allies: "What do you think *we* ought to try first? Why did you choose that? What will happen if *we* do this? How did you figure that out? What else could *we* do? Do you agree or disagree with that? Why? How else could *we* solve this? Is one way better than the others? How do you know?" When students make suggestions, the teacher must resist being the evaluator -- the answer-giver. Rather, the teacher should ask questions to explore the thinking and then test the student's suggestion on the computer. The computer will provide the feedback. "Why do you suppose that didn't work? What could *we* do differently next time? Why do you think that will make a difference? What's different about your two responses? What's the same? Which difference is critical? How do you know?" These questions help students to monitor their own thinking.

When the teacher assumes the role of guide rather than director, students experience attitudinal change -- they are more willing to take risks, to try different approaches. Risk taking is encouraged when the teacher's responses are characterized as accepting, building upon, integrating and extending student ideas. If risk taking is encouraged in the classroom, the *process* of learning will in fact produce learning.

How Does the Teacher Encourage the Transfer of Learning?

The computer can help create a manageable environment for focusing on skills. But at some point those skills need to be applied in a new environment if they are indeed to be learned. Transfer of learning from one situation to another is not automatic; it requires teacher direction. A critical step in effective teaching is to apply the skills or strategies taught in the computer-assisted lesson to other situations. For example, suppose the teacher has been using Sunburst's *The Incredible Laboratory* to focus on hypothesis formation and testing, and those same strategies apply in science units-- McGraw Hill's Elementary Science Study (ESS) "Mystery Powders," for example. It is the teacher's role to make the connection, to say, "Remember when we. . . ."

The computer has been derided by many as a passing fad, as a resurrection of teaching machines, as a toy, and as replacement for flash cards and workbooks. But the surge in development of software for teaching thinking skills suggests we may now have a way to teach skills we had not been able to teach effectively before computers.

Barry Beyer from George Mason University has related research on how we learn skills to the teaching of thinking skills specifically. Several of his findings are relevant to the teacher role discussed here. Beyer's research suggests that any skill is best learned when learners:

1. **Are consciously aware of what they are doing and how they do it.** The teacher-directed learning and questioning strategies recommended here help create that consciousness for learners.

2. **Are not distracted by competing input.** The microcosm created by the software eliminates distraction and focuses on specific skills and strategies.

3. **See the skill modeled.** The teacher, taking a collaborative position, can model the skill(s) relevant to the specific software.

4. **Verbalize their processes as they engage in the skill.** The "How do you know?" and "Why do you think that?" kinds of questions encourage this.

5. **Receive opportunities to practice the skill in contexts other than that in which the skill was introduced.** The teacher's efforts to facilitate transfer -- by relating skills learned to content area activities -- is critical for transfer of learning to occur.

Teaching thinking skills must be deliberate; we cannot expect such learning to occur by seating students at computers to independently experience even the best software. The teacher's role is the critical factor for capitalizing on the instructional potential of microcomputers. Remember: Teachers teach.

[Robert M. van Deusen, Grant Wood Area Education Agency, 4401 6th St. S.W., Cedar Rapids, IA 52404; and Jean Donham, Iowa City Community Schools, 509 S. Dubuque St., Iowa City, IA 52240.]

References

Beyer, Barry K. "Improving Thinking Skills -- Practical Approaches." *Phi Delta Kappan*, April, 1984, pp. 556-560.

Bondy, Elizabeth. "Thinking About Thinking." *Childhood Education*, March/April, 1984, pp. 234-238.

Burns, Marilyn. "The Role of Questioning." *Arithmetic Teacher*, February, 1985, pp. 14-16.

Readings (Number 2)

SOME ROLES OF COMPUTERS IN PRECOLLEGE MATHEMATICS EDUCATION IN THE 1990s: A FRAMEWORK FOR CHANGE

(This is a think piece paper prepared for the Task Force of the Mathematical Sciences Education Board of the National Research Council in the United States. The Task Force is doing long-range planning for computers in mathematics education. This paper was also presented and published in the Proceedings of the 1987 Alberta Teachers Association Computer Council conference. It has been revised several times since it was first written in 1986.)

David Moursund
International Council for Computers in Education
1787 Agate Street
Eugene, Oregon 97403

Introduction

The purpose of this paper is to provide a framework for planning major changes in mathematics education. Most educational leaders believe that our precollege mathematics education system in the United States is not as good as it should and could be. They cite as evidence test scores within this country, international comparisons, and a variety of national reports of study groups. They also point to computers both as a new challenge to mathematics education and as a possible aid in solving some of the problems of mathematics education.

Over the past 40 years, general-purpose electronic digital computers have become increasingly powerful and increasingly available. Computers have come into routine use by people who apply the mathematics they have learned in school. Computer assisted instruction has received considerable attention and has been extensively researched. It is evident that computers are useful aids to people who make use of mathematics.

Thus, two key questions have arisen:

1. Can computers help improve our (current, conventional) mathematics education system? This question focuses primarily on use of computers as an aid to learning mathematics.
2. Should the content of mathematics education be changed to reflect the availability and capability of computers? This question focuses on:

A. Use of computer-as-tool to help solve problems.

B. Changes in the curriculum content, such as increasing the emphasis on discrete mathematics, decreasing the emphasis on computation and symbol manipulation, and rearranging the order of presenting various topics.

This paper is divided into four major parts. The first discusses the hardware and software that one might base their long-range planning on. The second discusses the Accumulated Mathematical Knowledge of the human race, and how this is being changed by computer technology. The third gives a simple model of mathematical problem solving. When viewed in light of computer capabilities, this model suggests some changes in the mathematics curriculum. The fourth part makes recommendations based on earlier parts of the paper.

Computer Facilities

In planning for instructional use of computers in mathematics education, it is helpful to have some model of computer availability and capability in mind. The creation or selection of a model is a challenge, since both computer availability and capability are changing very rapidly. Almost every week one is apt to encounter news of a new product that is significantly better than the product it competes with. For example, a September 1986 issue of InfoWorld contained an article discussing a new 32-bit CPU chip that will be available by March of 1987 and that is twice as fast as currently available CPU chips.

Rapid progress is occurring in developing computer systems that make effective use of multiple CPUs. While the 256K-bit memory chip is now standard, mass production of the megabit chip has begun, and the four megabit chip is visible on the commercial horizon. Magnetic disk technology continues to make rapid progress and the optical disc is now a viable commercial product. Computer networks are being installed in schools and networking is becoming commonplace.

Rapid software progress is also occurring. We have new programming languages and improved operating systems, improved application programs, and integrated packages of applications software. In the past few years artificial intelligence progress has begun to lead to useful software products. The quality and quantity of computer assisted instructional materials continues to improve. CAI progress tends to be cumulative, and the newer materials tend to be more solidly based on both learning theory and careful formative evaluation.

Recent estimates are that there is about one microcomputer per 25-30 students in our precollege schools, and that this number is doubling every 14-18 months. A number of higher education institutions are requiring each student to own a microcomputer. It is already evident that over the long run access to computer hardware will not be a major limiting factor in instructional use of computers. People doing long-range planning for mathematics education should not dwell unduly on inadequacies of current computer capabilities and student access to these systems. Rather, they should assume that eventually every student will have easy access to a very powerful computer system.

For purposes of long range planning, we will describe a "Mathematics Education Computer System" (MECS) and assume that it will eventually be readily available to students as they study and use mathematics. The hardware of this system consists of:

1. 32-bit CPU(s) that can execute five to 10 million instructions per second (5-10 MIPS).
2. Primary storage of a minimum of two to four megabytes.
3. A high resolution, bit-mapped display with a resolution of approximately 1,000 by 1,000. This will be available in color where applications warrant the added expense.
4. Magnetic and optical disk secondary storage consisting of:
    A. A removable medium, such as 3-1/2 inch magnetic disks.

B. A hard disk system of adequate capacity to meet the demands of an educational environment.

C. Optical disc access, such as to a CD ROM system.

D. Hardware and software to access networks and remotely located databanks as needed.

It is evident that computer hardware systems will continue to improve rapidly over the next several decades. Twenty years from now the hardware for MECS will look modest relative to microcomputer systems being produced then. Indeed, it is likely that mass production will have begun on a single chip with the compute power and primary memory specified for MECS. But MECS has enough compute power to serve as a model for beginning major revision of our mathematics education system.

The software and courseware of MECS has three main components, all of which will continue to improve rapidly with time.

1. A mathematical reference library containing the equivalent of many hundreds of books. Materials would be available for students at a variety of grade levels and mathematical maturity levels. This library would also contain instructional support materials for teachers, such as back issues of the publications of the NCTM, sample lesson plans, courseware developed by federally supported projects, etc.

Note that one CD ROM disc can hold 550 million characters; a thick novel is about a million characters in length. A CD ROM can also store digitized pictures and diagrams. Thus, the above library can be stored on a modest number of CD ROMs. (The cost of making a large number of copies of a CD ROM, once an original has been produced, is under $2 each.)

One approach to storing reference books on CD ROM is to index them under every key word (excluding words like a, an, and, etc., which have little content meaning). But it is not at all clear that this is a good way to index mathematics reference books. This provides an excellent example of how the new technology is raising challenging problems for mathematics educators. Note also that our current mathematics education system does not give students much instruction in how to use mathematics reference materials.

Texts written specifically for access via computer can be interactive. They can make provisions for moving more deeply into a particular topic, or backing off and looking less deeply into parts of it. (Ted Nelson called this concept *hypertext* when he pioneered it in the early 1970s.) A whole new style of writing will need to be developed, along with a careful cross indexing system that helps guide readers through the wealth of available materials.

2. Applications (computer-as-tool). This would include a basic core of general-purpose applications software, such as a two-and-three-dimensional graphics package, a word processor designed to handle mathematical notation, a general purpose equation solver, a statistical package, a spreadsheet, a database, and an algebraic symbol manipulation system. It would also contain many hundreds of special-purpose programs designed to help solve more specific categories of mathematics problems. All of this software will need to be cross indexed with the reference materials discussed above and with the computer assisted instructional materials to be discussed next. Eventually all three of these sets of materials will need to be integrated into one comprehensive system.

3. Computer-assisted instruction materials covering the K-14 mathematics curriculum. In addition to traditional CAI, this would include simulations that create problem-solving environments, logic proof checkers, and other interactive aids to learning and doing problem solving.

It is evident that no precollege students currently have access to the MECS hardware and software we have described. However, many scientists and engineers have access to a combination of computer facilities, libraries, and support staff that are roughly equivalent to a significant portion of MECS. By 1995 many students will have access to a significant portion of this system. Moreover, mathematics education leaders could set a goal of making MECS available to all students.

Accumulated Mathematical Knowledge

Perhaps the single most important idea in problem solving is to build on the previous work of oneself and others. Mathematics, with its careful notation, precise definitions, and formal proofs, is well suited to helping people build on the previous work of themselves and others. A student learning to count and to write the numerals is building on the work of those who invented counting and the notation we now use for numerals. (For most purposes, it is a far superior notation than Roman numerals.) Students who have learned how to count can use this skill in solving a wide range of problems.

The accumulated mathematical knowledge of the human race is, roughly speaking, in three general categories of storage and processing media.

1. Human minds. Note that the human mind is both a *storage* and a *processing* medium.

2. Books, journals, written notes, photographs, paintings, and other passive media that can be repeatedly accessed. This category also includes phonograph records, tapes, movies, videotapes and other dynamic storage media that technological progress has produced in the past century.

3. Artifacts that people use to help do mathematics. This includes tools such as the abacus, sliderule, straight edge, compass, protractor, calculator, etc. Pencil, paper, chalk, and chalkboard can all be included in this category.

A protractor is an excellent example of a mathematical artifact. It embodies substantial mathematical knowledge. Most students can easily learn to make use of some of its capabilities to help solve problems. It is not necessary for a student to fully understand the mathematics embodied in a protractor, nor to understand all of its uses, to begin to make effective use of this tool. *A protractor, like many of the other mathematical artifacts, both stores mathematical knowledge and aids in processing or making use of the knowledge.*

A mathematics education system is designed to build on the capabilities and limitations of each of the three categories of storage and processing media. Any significant change to one of the categories may lead to a significant change in our mathematics educational system. For example, the development of reading and writing greatly changed Category 2 and certainly led to major changes in both the field of mathematics and in mathematics education. The development of movable type, another major change in Category 2 which eventually greatly increased access to books, changed mathematics education. In Category 3, solar-powered handheld calculators have had a significant impact on adults and a more modest impact on our mathematics education system. Gradually the use of calculators has come to be accepted in secondary school mathematics. But national assessments of mathematics still tend to prohibit use of calculators.

Computers impact each of the three storage media. First, consider the human mind. We now have very good research evidence that computer assisted instruction can help many students learn mathematics significantly faster and better as compared to traditional modes of instruction and/or as a supplement to these traditional modes. Moreover, complete courses can be delivered by CAI, providing good quality learning opportunities that might not otherwise be available to students. Finally, CAI allows increased individualization of instruction, with students working on materials appropriate to their levels and moving at paces appropriate to their abilities. CAI holds the potential to shift responsibility for learning mathematics more toward the student.

Category 2 consists of passive storage media such as books and dynamic storage such as phonograph records. It is evident that computers provide a new passive storage medium. Computers provide for the storage of a large amount of information in a small space. The previously mentioned CD ROM is just 14cm in diameter and the thickness of a phonograph record. But it can store 550 million characters--the equivalent of about 500 thick novels. Moreover, computer technology facilitates easy access to remotely located databanks. We are moving toward the time when the entire United States Library of Congress will be on line and readily available to people who need such access to information.

Computers provide a new type of dynamic storage, an interactive type of storage that is unlike anything we have had before. This is discussed later in this section of the paper.

Category 3, artifacts, contains tools that aid one in doing mathematics. We now have the possibility of students growing up with the computer tool. It seems evident that growing up in a MECS environment will shape students' minds in a manner quite a bit different from what occurs in a noncomputer environment. For example, consider computer graphics. Without computers it takes considerable effort and training for a student to represent data or functions graphically. Even a single, crude sketch of a function or a set of data can easily take minutes to produce. With MECS, graphing becomes a primitive that is usually accomplished in less than a second of computer time after the task has been specified. Students using MECS can create graphical representations of data with this tool at a younger age than they can without it.

Or, consider solving equations (polynomial, nonpolynomial, linear systems, nonlinear systems, etc.). The value of computers is obvious. Many time consuming and tedious tasks become primitives, routinely accomplished both rapidly and accurately by the computer, as one works to solve mathematics problems.

Or, consider linear programming and nonlinear programming. Students can learn to use these tools for mathematical modeling long before they can learn the underlying theory of solving such problems. Computers are already routinely used by all people who solve such problems.

The above analysis illustrates the most obvious ways in which computers impact the storage of accumulated mathematical knowledge. But there is still another, even more important idea. Computers represent a new, dynamic way to store some of the processes of applying human knowledge. In essence, a computer system is a medium combining all three storage categories. An application program designed to solve a particular category of problem both stores human knowledge on how to solve the problem and directs hardware to carry out the steps to solve the problem.

Research and development in artificial intelligence are gradually producing computer systems that capture some of the problem solving capabilities of human experts. Progress of this sort tends to be cumulative. Thus, more and more mathematical problems will be solvable by merely telling the problems to a computer. This topic deserves a much more detailed treatment than we can provide in the limitated space available here. Over the long run, progress in artificial intelligence may well change the basic nature of mathematics education. Students will grow up in an environment in which they learn to communicate with a computer system (by voice and keyboard) that has immense mathematical knowledge and ability to solve mathematical problems.

A Simple Model of Mathematical Problem Solving

In this section we present a simple-minded model of problem solving in mathematics. The purpose is to point out the main places where the MECS will impact people who use mathematics to solve problems.

1. Understand the problem. This may require making use of reference materials, and MECS will be useful. But to a large extent, understanding a problem requires drawing on one's total knowledge, asking probing questions, and interpreting problem situations in light of human values. It is a human endeavor, drawing heavily on the total interdisciplinary knowledge and skills of the problem solver. Often it requires good interpersonal communication skills.

A key point is that the typical real world mathematical problem is interdisciplinary in nature. One must know about the disciplines of the problem and about mathematics to understand such a typical real world problem. Currently many academic disciplines, such as the social studies, make minimal use of mathematics in their curricula. MECS provides tools that could change that. Increased application of mathematics throughout the school curriculum would make a significant contribution to mathematics education.

2. Develop a mathematical model of the problem. To a large extent, mathematical modeling is an intellectually challenging human endeavor drawing upon one's total knowledge of mathematics, the disciplines, the specific nature of the problem at hand, and one's experience in mathematical modeling. The MECS may be useful for information retrieval (for example, retrieving models that might be appropriate), drawing graphs and other pictures, word processing, etc.

MECS changes the range and nature of models available. Students can learn to use linear and nonlinear equation models, linear and nonlinear programming, etc., without knowing how to solve such problems using manual methods. Models can be used which require exhaustive searches of rather large solution spaces. Statistical models can be used which require extensive computations or exhaustive searches. Graphical models can be used, since two-and three-dimensional graphing is easily accomplished by computer. MECS has the compute power and graphical capability to do animation and color graphics.

3. Solve the mathematics problem developed in the previous step. Quite likely the MECS can do this or can make a significant contribution in doing this. Often this step is somewhat mechanical, and it is the step most conducive to being automated.

4. Interpret the results in light of the original problem. Return to Step 1 as needed. This mathematical unmodeling and interpretation process has the same characteristics as Step 2.

Even this simple model of mathematical problem solving makes clear that mathematics is and will remain a human endeavor. This model, and the discussion of the Accumulated Mathematical Knowledge, make it clear that one must know a lot of mathematics in order to do mathematics. But the doing of mathematics is highly dependent on the tools available and how well one has learned to use the tools. That is, learning to do mathematics is inextricably interwoven with learning to use the tools available to mathematicians.

Educators talk about a concept called *the teachable moment*. Imagine a person working to solve a mathematics problem but not having the knowledge and/or skills needed to solve the problem. We can imagine that the person might move from a problem solving mode into a CAI study mode to learn some aspect of the problem, and then back into a problem solving mode. This would be taking full advantage of a teachable moment. It represents a significant change in mathematics education which could help narrow the gap between learning mathematics and doing mathematics.

But there are two other key ideas evident from this simple model of problem solving. One is the idea of information retrieval. For many reasons we currently do a relatively poor job of helping students learn to use mathematics reference materials. The availability of MECS could (would) provide a strong incentive to make significant changes in this aspect of mathematics education.

The second major possible change comes from Step 3 above. Computers can execute algorithms quickly and accurately. One of the most mundane of examples is provided by a calculator and the issue of how much instructional time should be spent having students develop speed and accuracy at paper and pencil long division of multidigit numbers. The same arguments and ideas apply to graphing, solving systems of linear or nonlinear equations, doing statistical calculations, etc.

The basic nature of the human brain is that it is not good at doing repetitive tasks requiring extreme accuracy. It forgets, or becomes bored, or just plain makes an occassional error. The types of abilities that lead to excellence in doing repetitive computations or symbol manipulations seem only vaguely related to the higher-order, problem solving skills that we want students to gain through their mathematical studies. Indeed, it could well be that the emphasis on developing such skills is one of the roots of the "I can't do math and I don't like math" outcome that is so frequent in our mathematics education system.

The concept of an inverted curriculum has arisen from this type of analysis. In essence, the use of a computer to execute algorithms facilitates teaching students to use a computer to solve certain categories of problems without teaching them either the underlying theory or how to do the computations by hand. We currently have little research to help us understand posible effects of using a computer based inverted curriculum. But there are quite a few non-computer-based, somewhat analagous situations in our current curriculum.

The protractor was emphasized earlier in this paper because it illustrates some of the inverted curriculum ideas. Similarly, we teach grade school students to make use of a zero and a decimal point; both of these represented major breakthroughs in mathematics, and their underlying theories are well beyond students who are first learning their use. At the secondary school level we introduce the idea of a functoin and of functional notation. These are deep mathematical concepts, perhaps only fully understood by people who have both good mathematical ability and who study the subject for many years.

Note that cookbook statistics and other math-application courses existed before computers became available to students. Computers are increasing the possible range of such courses. Moreover, the MECS we have described is powerful enough to execute the artificial intelligence software that currently exists or is under development. More and more problems will be solvable by merely accurately specifying (describing) the problem to a computer. The computer will interact with the problem poser to assist in this accurate specification process.

Recommendations and Closing Comments

The basic recommendation is that mathematics educators and researchers work to create a MECS mathematics education environment for students. We have described a framework for change, and it can serve as a basis for long-range planning. The following five important steps need to be pursued concurrently and iteratively.

R1. Develop the hardware, software, and courseware of MECS and work to make the entire system cheaply and readily available to students. Begin orienting students to their responsibilities in a MECS learning and work environment.

(Note: As of May 1987 the Intel 80386 CPU chip and the Motorola 68020 CPU chip have both been in mass production for over a year; both provide adequate speed for MECS. The megabit chip has been in mass production for quite some time, and several companies have developed the 4 megabit chip. Some 20-megabyte hard disks retail for under $500. CD ROM systems have been available for nearly two years. Essentially all of the hardware components of MECS are in mass production for microcomputer systems. The Macintosh II with a math coprocessor chip and hard disk is roughly equivalent to MECS.)

R2. Provide appropriate training to exisiting and new teachers. This will require a massive amount of inservice training as well as changes to our teacher training programs. Increasingly, the role of CAI will change the role of teachers--perhaps to more of a mentor role.

R3. Begin both the development and the concurrent research on curriculum appropriate to a MECS environment.

R4. Begin modifying teacher-produced, district-wide, statewide, and national assessments to reflect and take advantage of a MECS environment.

R5. Begin working to gain the support of all of the people who must be involved in the changes needed to have mathematics education occur in a MECS environment. This includes students, parents, school board members, teachers, educational leaders, legislators, textbook publishers, etc.

We close this paper with a number of comments related to the ideas presented earlier. Many are points that require additional discussion and/or research.

C1. Computer facilities somewhat equivalent to MECS will increasingly become available to people in business, industry, government, and research. We know quite a bit about transfer of learning. We know that transfer of learning is greatly helped if the learning environment and the applications-of-learning environment are quite similar. This provides a strong argument for integrating the use of MECS into our mathematics education curriculum.

C2. Students vary widely in their mathematical abilities. Mathematics education is designed both to help students to work up to the levels of their mathematical abilities and to sort out those those with greater or lesser abilities. Those with greater abilities are encouraged to seek mathematically oriented careers, while those with lesser abilities are steered in other directions. But the sorting out process is often flawed. For example, students with poor ability to memorize computational and manipulative algorithms and to develop both speed and accuracy in their applications may be discouraged by our current mathematics education system, but we know that many such individuals have great mathematical ability. Education in a MECS environment might be of great help to people with low inate computational skills.

C3. Except in a few physical science courses, most nonmathematics courses make very little use of mathematics. The MECS tool has the potential to change this situation.

C4. For many people mathematics is a game to be played by certain rules. Thus, use of a calculator is cheating. It is evident that widespread availability and use of MECS changes the mathematics game. One can expect resistance to such changes. Quite a bit of the resistance will likely come from those currently playing the game quite successfully, such as math teachers. On the other hand, quite a bit of encouragement for the change may come from people who apply math on the job, such as scientists and engineers. For them, math is less a game and more an indispensible tool for solving the problems they encounter on the job.

C5. Our mathematics education system is used to tools such as the compass and protractor. Such tools change very slowly, if at all, during a person's lifetime. Our mathematics education system is not used to rapidly changing tools. Mathematics education, especially at the precollege level, is built on content that may change little during a person's teaching career, and on methodology that changes but little over several decades. Thus, our mathematics education system is basically conservative in nature. This suggests that it will be quite difficult to move this system in the direction of the MECS environment.

C6. Color displays and motion graphics add new dimensions to the tools available to students and teachers. We know little about appropriate uses of such tools. Research is needed.

C7. We have avoided comment on the teaching of computer programming and computer science. These are topics that are related to change in mathematics education, but are not at its core. Computer science is a discipline that is distinct from mathematics. However, mathematics educators may decide that it is advantageous for all mathematics students to learn to program. Current trends suggest that if this is to occur, it will likely be at the college level. However, there is little indication at the college level that this will occur on a widespread basis.

C8. The ideas proposed in this paper will require many decades to implement. But a significant start can occur in the next 10 years. The microcomputers currently available in schools are powerful enough to begin the change to a MECS mathematics education environment.

C9. MECS, and the ideas discussed in this paper, could revitalize mathematics education. It could bring new life and excitement to mathematics students, faculty, researchers, and writers.

Appendix

Problem Solving:
an Annotated Bibliography

by
Billy C. Yates

There are several thousand articles, papers, and books on problem solving. Even a limited review of the literature is difficult, and the field is changing fairly rapidly. Most of the entries in this bibliography are from the last few years of published material. This relatively small sample represents a large number of ideas and research findings.

Most of the articles are written by experts in the fields of science, psychology or education. Some research papers were selected because they show strong experimental results or support a common problem solving theme in the literature. Books, edited by experts in their fields, were included because they represent current thinking about problem solving in their discipline. Other books are included because they represent an alternative view or a philosophical or historical tradition in problem solving.

Humans are incredibly complex organisms, and many thinkers have said that the small mass that makes up the human brain is at the apex of that complexity. The diversity of our problem solving behaviors is representative of that underlying sophistication. To study this most complex of all behaviors is a formidable task, but there has been over the last few years a renewed interest in problem solving, and some interesting opinions, facts and conclusions have been expressed.

For example, the following is a list of findings gleaned from some of the research.

(1) Putting a problem aside and letting it incubate, contrary to the common view, may not facilitate it's solution.

(2) Problem solvers who talk about the steps they are taking to solve the problem do better than those who do not describe their efforts.

(3) Problem solving skills used in groups do not necessarily transfer to individual problem solving skills.

(4) Even with well-structured problems, people tend to frame small subgoals and may not be able to explain why they do so.

(5) How we think about (or represent) a problem is a better indicator of the problem's difficulty than is any quality intrinsic to the logic of the problem.

(6) We seem to have a few basic, general problem solving strategies for working with a variety of problem situations.

(7) Good problem solvers have better mental representations of the problem than less able problem solvers.

(8) Precise thinking (processing) is one of the keys to strong problem solving ability.

(9) How conscious we are of our thinking processes while solving a problem is dependent on whether we are using a familiar strategy or having to develop the strategy as we work.

(10) Changing our perspective on a problem often aids in arriving at a solution.

(11) Experts, out of their domain of expertise, do no better than novices.

(12) On certain types of problems, the computer has been shown to be a strong problem solving aid.

(13) Groups made up of two to four individuals using the computer are better at problem solving than either individuals or groups composed of five or more people.

(14) Good problem solving ability in one area does not carry over to problems in another area.

(15) The more you know, the better problem solver you tend to be.

(16) With experience we improve as problem solvers, but we seem to have difficulty transferring that knowledge to like or analogous problems.

The research indicates there are many practical steps that can be taken to help students become better problem solvers. Although there is no consensus on a general problem solving strategy that will work in all situations, some of the facts in the above list can be of use to the classroom teacher.

There is every indication that problem solving is domain specific. That is, the skills learned in one discipline do not easily transfer to another. While this may be true, some educators indicate that if we teach an integrated approach to problem solving in the classroom and throughout the curriculum then we will provide a diverse problem solving environment and presumably develop more skilled problem solvers as a result.

Psychologists believe that the brain is stimulated by a rich environment, and a diverse approach to problem solving is encouraged.

The two hemispheres of our brain are specialized in slightly different ways, and some educators believe that a balanced program that includes problems emphasizing the use of each hemisphere is needed. Some philosophers have complemented this view by encouraging us to teach everyday ill-defined problem solving as well as more formal processes based on logic.

There are many theories about human intelligence. Some reseachers view us as having three broad categories of intelligence and others think there are seven or more basic categories. Likewise the information processing view of intelligence, represented by the cognitive scientist, has also taken a number of different forms. But not withstanding the diverse views about human intelligence, most all of the researchers recognize that intelligence is strongly related to problem solving ability.

A number of educators believe the computer can play a very active role in teaching problem solving. According to some authors, databases and other programs are helpful teacher/student tools in experiencing and learning about problems. Some curricular programs have been built around the computer, and the results, while not dramatic, indicate that the use of the computer is a plus not only for developing higher-order thinking skills, but for learning basic skills as well.

We know something about problem solving, but much of what is known is taken from problems that are well defined. These problems have clearly defined starting states and goals. Little research has been done on in-context (real world) problems that do not have clearly defined starting and ending states. This research is supplemented by a rich tradition of anecdotal observations. Taken as a whole, the literature can serve as a general guide in teaching problem solving skills, but no clear, proven theories have emerged concerning the human problem solver.

All the material in this bibliography is annotated. The entries are not intended to be comprehensive reviews, nor do they include all the important material in a reference. They summarize, in a broad sense, the articles, papers and books. A real understanding of the authors' ideas can be gotten only by reading the actual works.

Anderson, J. R. (Ed.). (1981). *Cognitive Skills and their Acquisition.* Hillsdale, NJ: Erlbaum.

This book contains a number of articles by various writers on learning and skill acquisition. One essay by Anderson, Greeno et al deals with the acquisition of problem solving skills. This is a technical paper about students learning to solve geometry proof problems. The authors suggest that three stages are involved in gaining expertise in problem solving. They are: (1) the declarative stage, when the learner places in short-term memory facts about the skill; (2) a compilation stage, when the learner begins to internalize the procedure and can carry it out; and (3) a procedural stage, in which the learner can, with little conscious effort, solve the problem. Like chess players, experts at proofs in geometry have simply encoded many special case rules (solution examples).

Baker, G. E., & Dugger, J. C. (1986). "Helping Students Develop Problem Solving Skills". *The Technology Teacher,* (January), pp. 10-13.

This article draws a parallel between the scientific method and problem solving strategies recommended for the industrial arts classroom. The five steps discussed are: (1) set an attainable goal for the student, (2) define a task that will require the student to use new methods or ideas, (3) provide a structure so the student can test the consequences of his or her actions, (4) force the student to choose amoung several alternatives, and (5) ensure that the student evaluates his or her work to see if the ideas worked or failed and how to improve the problem solving process. The authors emphasize that just providing problem solving activities is not enough; the teacher much provide structured activities for the student to receive the maximum benefit.

Bender, T. A. (1986). "Monitoring and the Transfer of Individual Problem Solving." *Contemporary Educational Psychology,* (April), pp. 161-69.

This research paper requires careful reading. The authors conclude that when problem solvers verbalize as they solve a problem, they become better able to solve problems in other situations. The study adds to the already-large body of research that indicates that initial group superiority in problem solving is generally not transferred to the individual. That is, group problem solving exercises do not necessarily improve individual problem solving skills.

Berry, D. C. (1983). "Metacognitive Experience and Transfer of Logical Reasoning." *The Quarterly Journal of Experimental Psychology, 35A*, pp. 39-49.

This is a research paper studying the process of transfer of problem solving skills from concrete to more abstract problems. While solving problems, those experimental subjects who were allowed to talk about how they were thinking through their solution did better on more abstract tasks than those who were not allowed to talk as they were solving similar problems.

Best, J. B., *Cognitive Psychology*. New York: West Publishing Company .

This readable textbook provides a good survey of the relatively new field of cognitive psychology, the study of cognitive (mental) processes. The book covers the general areas of perception, memory, language, and thinking in people. It has a chapter devoted to problem solving. This chapter looks at the information-processing approach to problem solving in contrast to the Gestalt tradition. The experimental evidence supporting both theories are examined. The information-processing approach appears to have the greater experimental support.

Beyer, B. K. (1983). "Common Sense About Teaching Thinking Skills." *Educational Leadership, 41*, (November), pp. 44-49.

Beyer gives some practical advice to educators about the teaching of thinking skills across the curriculum. The guidelines are: (1) a supportive learning environment which teaches skills involving comparisons, analysis, invention, and inference is essential; (2) using effective strategies which involve teaching thinking skills over a period of time that not only introduce the skill but practice and reinforce it; (3) clear curriculum guidelines for teachers (The curriculum guidelines will clearly define the skills for the teacher. It is suggested that the school introduce three to five thinking skills per grade level and relate thinking skills to be taught in one content area to those taught in another using various media); and (4) support the teaching of thinking skills across the curriculum.

Beyer, B. K. (1984). "Improving Thinking Skills -- Defining the Problem." *Phi Delta Kappan*, (March), pp. 486-490.

This is the first of two related articles . The author outlines several major obstacles to improving thinking skills: (1) educators have no consensus about what is meant by thinking, (2) we have failed to identify those cognitive operations that make up thinking skills, (3) classroom teachers do not provide concrete instruction on how to exercise a particular thinking skill, (4) there is little time to teach the incredible variety of skills already taught, and (5) we use tests that measure discrete skills in isolation and think that tests are the only way to measure thinking skills in students.

Beyer, B. K. (1984). "Improving Thinking Skills -- Practical Approaches." *Phi Delta Kappan*, (April), pp. 556-560.

This is the second half of Beyer's series addressing the improvement of students' thinking skills. Beyer gives several practical suggestions to improve thinking skills. The topical areas are: (1) identify the specific skills we want to teach, (2) carefully conceptualize the skill components we are to teach, (3) provide appropriate instruction by taking into account the research on the teaching of thinking skills, (4) develop a curriculum that is coherent (see Beyer 1983), and (5) improve the tests used to measure thinking skills.

Bolter, J. D. (1984). *Turing's Man: Western Culture in the Computer Age.* Chapel Hill: The University of North Carolina Press.

While the author does not directly look at problem solving, he does address the impact of the computer on our society from a radically different perspective than the cognitive scientist's view. In each age people have assigned to the machine the capability to redefine their notions of people and space. The computer is no different, but the author suggests that, notwithstanding the simplistic and historically naive view that many information theorists hold, the computer is a useful tool. In sum, the author argues for a "synthetic intelligence," as distinct from "artificial intelligence," that represents a synthesis between people and the computer.

Carey, M., Foxman, P. N., Tarr, D. B., "Verbalization, Experimenter Presence, and Problem Solving." *Journal of Personality and Social Psychology, 8,* pp. 299-302.

Studies indicate that verbalization during problem solving facilitates performance. This study examines what effect the experimenter has on such results. While the experimenter's presence may influence the subjects performance, the effect is not sufficiently large or consistent between different tasks to indicate undue influence. The authors conclude that the presence of others is a complicated variable and much more study is warranted before any strong conclusions can be reached.

Cox, D. A. & Berger, C. F. (1985). "The Importance of Group Size in the Use of Problem-Solving Skills on a Microcomputer." *Journal of Educational Computing Research, 1,* pp. 459-468.

This study was performed to find out whether group size is related to microcomputer problem solving success and whether computer use reduces the time necessary to solve problems. The researchers used groups of two, three, four, or five middle school students. The students in groups of two, three and four solved more problems than the students in groups of one or five. Groups of five solved problems as well as other groups, but there tended to be more social friction in the group. The study recommends that students using computers be placed in groups of two, three, or four members.

Eccles, J. (Ed.). (1987). *Mind and Brain: The Many-Faceted Problems.* New York: Paragon House Publishers.

This book, edited by nobel laureate Eccles, is a collection of articles addressing the philosophical issues concerning the relationship between the mind and the brain. This is an important work for anyone who considers questions about human consciousness and the philosophical roots that determine our view of that consciousness. The essays address perhaps the quintessential problem to be solved: people's inner mental nature. Some very deep issues are addressed

in this book, and they require a considered reading.

Eiser, L. (1986). "Problem-Solving Software: What it Really Teaches." *Classroom Computer Learning*, (March), pp. 42-45.

The author suggests that students should be given a set of mental guidelines that will help them find a quick and efficient solution to problems. She further contends that computer software can facilitate the teaching of a wide assortment of thinking skills. A wide variety of computer software is reviewed with respect to the relevant thinking skills, such as memorization and recall, noting patterns, breaking a task into parts, trial and error experimentation, note taking, and drawing conclusions from clues. Software purchase information is supplied (such as vendors' addresses) as is a chart that lists the various thinking skills that can be practiced with various problem problem solving software.

Fischler, M. A., & Firschein, O. (1987). *Intelligence: The Eye, the Brain and the Computer.* Reading, MA: Addison-Wesley Publishing Company.

This is a readable but technical book that "is intended to be an intellectual journey into the domain of human and machine intelligence." The authors look at learning by humans and machines. They include in the chapter on reasoning and problem solving some interesting studies on human logical reasoning. Experiments have shown that we learn and use rules of inference such as, "If John is good, he will be rewarded. John is good. Therefore, John will be rewarded." But we are not comfortable with "If John is good, he will be rewarded. John is not good. Therefore, John is not rewarded." Apparently we tend not to think in the negative. They cite another study which indicates that how we think about (or represent) a problem is a better indicator of the problem's difficulty than any quality intrinsic to the logic of the problem. Through these and other examples, the authors point out some of the major differences between everyday human problem solving methods and formal (logical) reasoning processes.

Forbes, R. H. (1984). "Thinking Skills: What Are They? Can They Be Taught? Why and How?" *National Association of Secondary School Principals Bulletin, 68,* (December), pp. 68-75.

This article addresses the shift from an industrial work force to a high tech work force and what that shift means to educators. Forbes contends that there are a few high tech jobs requiring higher-order thinking skills, but most new jobs require the use of technology at a nontechnical level. Thus, the ability to advance in an organization by enterprise is quickly disappearing. He encourages us to teach those skills that the graduate will need to compete for better paying jobs and those skills that allow the graduate to live in a technological society. This requires schools of education to explicitly teach teachers how to teach thinking skills .

Frederiksen, N. (1984). "Implications of Cognitive Theory for Instruction in Problem Solving." *Review of Educational Research, 54,* pp. 363-407.

This is a comprehensive article that reviews various cognitive theories of problem solving and looks at possible applications to the classroom. Most theories of problem solving rest on information processing concepts and are applied to well-defined problems and problems requiring rule-based thinking. General methods (heuristics) of problem solution are looked at, and contrasting views are summarized. Cognitive psychologists suggest what processes to teach (problem structure, pattern recognition, knowledge base, knowledge structures, and specific thinking skills), but not how to teach them.

Gagne, R. M. & Smith, E. C. (1962). "A Study of the Effects of Verbalization on Problem Solving." *Journal of Experimental Psychology, 63,* pp. 12-18.

This study was conducted to determine the effects of verbalization on problem solving ability. The subjects were freshman and sophomore high school students. The students who talked about how they were attempting to solve the problem solved problems better than those who did not verbalize their thinking processes. The people who were required to verbalize their solutions also solved the problems in a shorter period of time. The authors suggest that verbalization has the effect of making the students examine alternative lines of reasoning, thus facilitating the discovery of general rules and the application of these rules to like problems.

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

This book is a major departure from the traditional IQ concept as a measure of intelligence. Intelligence as viewed by the author consists of seven components: (1) linguistic intelligence, (2) musical intelligence, (3) logical-mathematical intelligence, (4) spatial intelligence, (5) bodily-kinesthetic intelligence, and (6) two personal intelligences (*intra*personal, *inter*personal). Gardner says, "The world is enwrapped in meanings, and intelligences can be implemented only to the extent that they partake of these meanings, that they enable the individual to develop into a functioning, symbol-using member of the community."

Greeno, J. G. (1976). "Indefinite Goals in Well-Structured Problems." *Psychological Review, 83,* pp. 479-491.

The author has found that when people solve a well defined problem (see Moursund), such as proving a theorem in high school geometry, they develop numerous sub-goals that are not always well defined (indefinite goals). In *problem situations* an individual typically sets many sub-goals without knowing how the information obtained will

be used. But this phenomenon has not been well documented in well defined problems. The author suggests this finding should provide a modest extension of our problem solving theories concerning well-defined problems.

Groner, R., Groner, M. & Bischof, W. F. (Eds.). (1983). *Methods of Heuristics*. Hillsdale, NJ: Lawrence Erlbaum Associates.

This is a collection of essays concerning the general methods of problem solving. It contains articles on heuristic (general) problem solving skills by psychologists, mathematicians, and computer scientists. This interdisciplinary approach provides interesting but technical reading. The essays generally attempt to exemplify the methods used in problem resolution in each discipline. The editors conclude that there is no agreement on a definition of *heuristic* across disciplines, nor whether, such a definition is even possible.

Haugeland, J. (1985). *Artificial Intelligence: The Very Idea*. Cambridge: A Bradford Book. MIT Press.

This book addresses the issue of artificial intelligence. But to the uninitiated it will require more than one reading. The author outlines how we viewed the mind historically and how we have now come to view the mind. He then takes the reader through formal systems (games with special rules) and the nature of meaning in mechanical systems and the architecture of computer systems. He then looks at practical examples of artificial intelligence. He ends with a comparison between the computer and the human condition and questions whether human qualities can be encoded into a machine. An interesting book well worth reading.

Hintzman, D. L. (1978). *The Psychology of Learning and Memory*. San Francisco: W. H. Freeman and Company.

This textbook takes a rigorous experimental approach to learning and memory. It gives an excellent historical perspective on the development of psychology from the early introspectionists through classical, instrumental conditioning and cognitive psychology. While problem solving is not directly addressed in the book, much of the work described is supportive of many of the recognized general problem solving methods.

Hunter, B. (1985). "Problem Solving with Databases." *The Computing Teacher*, (May), pp. 20-27.

This is a descriptive paper. It covers topics such as: (1) what a data management program is; (2) why teachers and students use data bases, (3) some example files, (4) stages of learning using data bases, supported with examples and activities at each stage; and (5) how to select data management tools for the classroom. The author believes that databases provide a rich source of activities for student inquiry, problem solving and skill development in subject areas. The resource is there, she asserts, and the quality is improving rapidly.

Kellogg, R. T. (1982). "When Can We Introspect Accurately About Mental Processes?" *Memory and Cognition, 10,* pp. 141-144.

Some psychologists have argued that when we introspect and verbalize how we are solving a problem, we give an accurate portrayal of how we are solving a problem. Others have suggested that much of what is involved is unconscious, and therefore verbalizations are not accurate descriptions of a problem solving process. This study gives evidence that both views may be true based upon the type of task involved. If the problem requires conscious effort, verbalizing may provide an accurate picture. But if the task uses automatic procedures (previously learned problem solving methods), then the use of these procedures may not be reflected in the verbal reports.

Kleinmuntz, B. (Ed.). (1966). *Problem Solving: Research, Method, and Theory.* New York: John Wiley & Sons, Inc.

This is the collection of papers presented by psychologists at the first annual symposium on cognition held at the Carnegie Institute of Technology during April 1965. The papers reflect the view of problem solving in the mid 1960s, which is one decade past, (some suggest) the decade that marks the modern interest in problem solving. The papers are a representative sample of two problem solving traditions: the learning-psychology group (laboratory and experimental approach to learning and problem solving) and the information-processing camp (computer models can provide theoretical models of human thinking).

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

The authors examine how experts solve problems in physics. Their view is that all experts use the same basic processing characteristics independent of the subject matter within which the expert works. Experts typically use an often-exercised body of knowledge that is indexed by a large number of patterns or successful routines developed through experience. This experience guides the expert to solutions that we ordinarily interpret as physical intuition. Now that computer systems can simulate expert like processing, the intuition (pattern-indexed schema) used by experts should no longer be mysterious or inexplicable.

Lipman, M. (1984). "The Cultivation of Reasoning Through Philosophy." *Educational Leadership, 42*, (September), pp. 51-56.

Lipman believes that the one major contribution to problem solving that can be made by philosophy is teaching ". . .thinking that devotes itself to the improvement of thinking." Higher-order thinking skills are just combinations of simple logical relationships, and these skill can be taught through stories and cooperative discussions of the themes in the stories. It is not enough to be conversant with the content of the subject; the student must be made to think in the subject. The guiding discipline of philosophy can and should be used to teach school children the difference between good thinking processes and not-so-good thinking processes. He suggests that a series of philosophy courses, with appropriate support materials, should be introduced into the curriculum (K-12).

McAuliffe, K. (1985). "Making of a Mind." *OMNI*, (October), pp. 63-66.

This article examines the developmental stages of children and their sensitive periods (optimal periods for certain forms of learning). For example infants, until the age of one year, can hear all the sounds in any foreign language (such as some sounds unique to Hindi), but after that they lose the ability to hear certain sounds if those sounds are not sustained in their native language. Other examples are given from other human and animal studies. In essence the brain is a very pliable organ, and a rich environment optimizes its ability to perceive and handle information.

Moursund, D. (1986). *Computers and Problem Solving: A Workshop for Educators*. Eugene, Oregon: International Council for Computers in Education.

This is a book directed at educators. A definition of a well-defined problem is given and carefully demonstrated. This concept is supplemented with other practical examples that demonstrate many of the key ideas in problem solving. The roles of proven recipes (algorithms) and general guidelines (heuristics) in solving general categories of problems are discussed along with the computer as a problem solving aid or tool. The ideas in the book are clearly stated and supported with practical examples. This book is well worth reading and studying.

Nickerson, R. S. (1984). "Kinds of Thinking Taught in Current Programs." *Education Leadership, 42*, (September), pp. 26-36.

This article looks at the teaching of thinking skills in the school. It lends credence to explicitly teaching thinking skills within the curriculum. Using research to support the major ideas, the author looks at: (1) examples of programs to teach thinking, (2) cognitive processes, (3) heuristics, (4) formal thinking or stage development, (5) language and symbol manipulation, (6) thinking about thinking, (7) learning strategies, and (8) thinking skills and knowledge. There appears to be no one model for teaching thinking skills that stands out. While there is no clear evidence that these methods significantly improve student performance, the author suggests the goal is reasonable and general progress is being made in this area.

Olton, R. M. & Johnson, D. M. (1976). "Mechanisms of Incubation in

Creative Problem Solving." *American Journal of Psychology, 89,* (December), pp. 617-630.

Many people believe that setting a problem aside for a while to incubate can be helpful in arriving at a solution. This research article has demonstrated that those who took time to think about other things and then go back to the problem did not do as well as those who were not given time to let the problem incubate. The author reviews other research on the role of incubation in problem solving and concludes that no one has been able to replicate a single research effort where someone has indicated that incubation facilitated a problem solution.

Paul, R. W. (1984). "Critical Thinking: Fundamental to Education for a Free Society." *Education Leadership, 42,* (September), pp. 4-14.

The current view is that problem solving should be taught explicitly and tacked on to other learning. This view, called by Paul a "short-term strategy," is incomplete and leads to a narrow form of technical knowledge. A better long-term strategy is to teach students to think critically and reciprocally within different or opposing points of view (dialectically). Only by principled, not procedural, thought can students be taught comprehensive rational principles that lead to a more genuine democracy of thought.

Pogrow, S. (1985). "Helping Students to Become Thinkers." *Electronic Learning,* (April), pp. 26-29, 79.

Pogrow describes how he and other teachers use the computer and related software to teach higher-order thinking skills (HOTS) in grades four through six. Methods of computer use by students and teachers are described, and the results of the project indicate that higher order thinking skills as well as basic skills of students in these grades are improved. The author believes the project demonstrates that drill and practice and exercises that develop higher-order thinking skills are both legitimate uses of the computer.

Reed, S. K., Ernst, G. W. & Banerji, R. (1974). "The Role of Analogy in

Transfer Between Similar Problem States." *Cognitive Psychology, 6*, pp. 436-450.

Problem solving improves with experience. The question asked by the authors of this research study is, "How much does solving one problem help in solving another problem of similar type?" They conclude that, without help, there is no substantive transfer of skills used to solve the first problem to the solution of the second problem. Transfer of skill between problems *does* occur when a subject solves the same problem again. However, when subjects were given the suggestion that the problems had similar characteristics, the results indicated that some transfer occurred, but the transfer was not dramatic enough to warrant drawing unambiguous conclusions.

Royer, J. M. (1979). "Theories of the Transfer of Learning." *Educational Psychologist, 14*, pp. 53-69.

This is a technical paper that examines the issue of transfer of learning from one context or situation to another. Two broad categories or theories of transfer are examined with the relative strengths and weakness of each explained. The environmental theory, based on the behavioristic tradition and embodied in programs such as *Distar*, appears to provide guidelines for developing and sequencing instructional events. In contrast, the cognitive psychologists' theories give no specific guidelines for the educator but may provide general suggestions for providing transfer from school-related skills to real world situations (such as teaching by analogy).

Samples, B. (1978). *The Metaphoric Mind: A Celebration of Creative Consciousness.* Menlo Park, CA: Addison-Wesley.

This is a unique and beautifully written book. The right hemisphere of the brain tends to be creative and metaphorical whereas the rational left hemisphere tends toward the logical and analytical. Harmonizing these two tendencies brings a new, synergetic unity to our lives. Samples gives us his vision when he writes, "The Parent -

nature and its mind of metaphor - is quietly, lovingly urging the child - culture and its mind of rationality - that the time has come to celebrate unity . . . to live the oneness destined by time's formless course."

Scheerer, M. (1963). "Problem-Solving." *Scientific American, 208,* (April), pp. 118-128.

Scheerer has found that when solving a problem you can and should change your approach. Several experiments have shown that when problem solvers becomes fixated with one approach to the problem, the solution processes is inhibited. Subjects who changed their perspective and tried a different approach arrived at an answer quicker than those that persistent in only one approach. Often sudden shifts in perspective facilitate problem solutions.

Simon, H. A. & Simon P. A. (1962). "Trial and Error Search in Solving Difficult Problems: Evidence From the Game of Chess." *Behavioral Science, 7,* pp. 425-429.

This technical article looks at how world-class chess players go about playing championship chess. Simon and Simon suggest that the prodigious memories and processing capabilities of outstanding chess players is a myth. To examine all the moves of a given chess position requires over a billion combinatorial views. To examine all these combinations would be an amazing mental feat, but it seems that grand-master chess players do not review all the possible moves. They use a powerful selective general strategy (outlined in this article). Thus, grand-masters do not need prodigious memories or very fast processes capacities, but through practice develop a very selective but effective problem solving strategy.

Speedie, S. M., Treffinger, D. J. & Houtz, J. C. (1976). "Classification and Evaluation of Problem-Solving Tasks." *Contemporary Educational Psychology, 1,* pp. 52-75.

The authors take the position that all problem solving tasks can be described using three general categories: (1) task environment characteristics, (2) classes or processes involved in problem-solving tasks, and (3) the measures yielded by the tasks. Speedie et al. believe their classification scheme has at least two principal implications for educators: (1) describing and documenting goals without a unified model is difficult, and their scheme provides a conduit for a more precise and effective definition of instructional goals and objectives; and (2) the educator may use this scheme to select tasks to assess the accomplishment of specific goals in the problem solving domain.

Steinberg, E. R., Baskin, A. B. & Hofer, E. (1986). "Organizational/Memory Tools: A Technique for Improving Problem Solving Skills." *Journal of Educational Computing Research, 2* (2), pp. 169-87.

This is an experimental study that uses the computer as a memory aid to solve a class of problems called "functional dependency networks." These networks have many parts, and each part is dependent on several others. If one part fails, this failure generates other failures, and so on, until the entire system collapses. Solving problems using these networks requires keeping track of a large number of details about the system. The computer was used as an aid to assist students in remembering those details. While it took some time to set the computer up to serve as a memory aid, the students using the computer did as well as those who did not. This suggests that if you factor out the time used to learn how to apply the tool, the computer users did better than those without such an aid.

Sternberg, R. J. & Davidson, J. E. (1982). "The Mind of the Puzzler." *Psychology Today,* (June), pp. 37-44.

This article focuses on the importance of insight in problem solving. The authors identify three types of insight: the ability to select relevant information; the ability to combine different seemingly unrelated information in a useful form; and the ability to compare the current problem with like problems previously encountered. But apparently these qualities alone are not enough. The authors conclude that problem solving also involves a number of other abilities, including prior knowledge of similar situations or congruent facts; executive processes used to plan, monitor, and

evaluate performance; and cognitive style (the impulsive versus the reflective personality, for example).

Sylwester, R. (1985). "Research on Memory: Major Discoveries, Major Educational Challenges." *Educational Leadership, 42,* (April), pp. 69-75.

This article relates our knowledge about memory to the school curriculum. The results of research on procedural (learned but now automatic skills like driving) and declarative (factual) memory suggest several curricular challenges: what knowledge and skills students should commit to memory; how we can help students move from memorizing random facts to creating useful concepts; how we can teach students to use memory effectively in problem solving; and how we can use neuroscience research to enhance the effectiveness of student memory. The author uses current research to speak to each question and encourages educators to "get into the literature and to begin to think about the curricular challenges that will most certainly emerge."

Sylwester, R. (1986). "Synthesis of Research on Brain Plasticity: The Classroom Environment and Curriculum Enrichment." *Educational Leadership, 44,* (September), pp. 90-93.

This very readable article looks at the enriched environment research (the effect of a rich environment on the development of the neocortex) and the educational questions this research raises. Concerning the brain's development, Sylwester states, "The basic development pattern is: (1) create an initial excess of connections (axons/dendrites/synapses) among related areas, (2) use learning and experience to strengthen the useful connections and then prune away the unused and inefficient, and (3) maintain enough synaptic flexibility to allow neural connections to shift about throughout life as conditions change and new learning/problem solving challenges emerge." He further asserts, "the challenge to educators is simple: define, create, and maintain an emotionally and intellectually stimulating school environment and curriculum." It is too early to get specific guidelines from the research, but the research is beginning to support the long-held tenet among educators that an enriched educational environment positively affects students'

development.

Trotter, R. J. (1986). "Profile Robert J. Sternberg: Three Heads Are Better Than One." *Psychology Today*, (August), pp. 56-62.

This is an excellent summary of the work of Robert J. Sternberg and his triarchic theory of intelligence. Basically, Sternberg suggests that we are governed by three aspects of intelligence: (1) componential (good at tests and analytical thinking), (2) experiential (creative thinker), and (3) contextual (adaptive and street smart).

Whimbey, A. (1984). "The Key to Higher-Order Thinking is Precise Processing." *Educational Leadership, 42*, (September), pp. 66-70.

This paper reviews the research of Bloom and others and concludes that the difference between good problem solving styles and poor problem solving styles is in the depth of processing. Good problem solving students tend to employ a lengthy sequential analysis in arriving at an answer, whereas low-aptitude students tend to take a very superficial or one-shot thinking approach to a problem. Precise processing is the key attribute of higher-order thinking. This should suggest to the educator that we not change educational material but rather place more emphasis on mental processing and make provisions to observe and guide students with feedback on their own processing.