DOCUMENT RESUME

ED 244 593 IR 011 114

AUTHOR	Walker, Decker F.
TITLE	Computers in Schools. I: Potential & Limitations. II: The Software Problem. Educational Brief.
INSTITUTION	Far West Lab. for Educational Research and Development, San Francisco, Calif.
PUB DATE	Feb 84
NOTE	13p.; Prepared by the Improvement Support Program.
PUB TYPE	Viewpoints (120)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Software; *Educational Improvement; Elementary Secondary Education; *Microcomputers; Problems
IDENTIFIERS	*Computer Uses in Education

ABSTRACT

Some ways in which microcomputers can contribute to the quality of instruction are outlined and problems that limit their usefulness, particularly the scarcity of good software, are discussed. The advantages of using microcomputers as instructional tools are identified as more active learning, more varied sensory and conceptual modes, learning with less mental drudgery, learning nearer the speed of thought, individually tailored learning, more independent learning, and better aids to abstraction. The following seven problems of using microcomputers are also outlined: (1) the fact that microcomputers can supplement but not substitute for conventional education; (2) difficulties in their use; (3) rapid change and lack of standardization in the microcomputer marketplace; (4) the scarcity of good software; (5) the lack of knowledge on optimum methods of educational computing; (6) the fact that computers favor formalism over judgment; and (7) the inability of computers to solve social and other problems in education. Software development time and cost, machine incompatibility, software piracy, difficulties in software location and review, home market competition, and the difficulty of integrating software into other classroom activities are identified as factors contributing to the software shortage problem. Suggestions for overcoming the software problem are also briefly reviewed, such as the use of tool-type, modifiable, and electronic blackboard programs. (ESR)

EDUCATION BRIEF

February 1984

COMPUTERS IN SCHOOLS

I: POTENTIAL & LIMITATIONS
II: THE SOFTWARE PROBLEM

Decker F. Walker
School of Education
Stanford University

The romance and adventure the personal computer holds for many educators and laymen are undeniable. In a time of stringent fiscal austerity schools still find ways to buy computers for the classroom. PTAs collect money through bakesales, parents donate equipment, teachers buy machines for their classrooms out of their own funds, principals find ways to economize on toilet paper and use the savings for a computer. Both the public press and professional journals sport articles on computers in nearly every issue.

It is impossible to predict how long computers-in-education will hold the spotlight of public and professional attention. Five years would be an excellent run; 10 years would be the best since progressive education. What concerns me, however, is not the length of the run but what will remain in the schools when the boom has run its course and a new act has captured the spotlight. When that happens, the fate of computers-in-education will depend on the instructional uses to which teachers and students can put their computers.

This paper will outline some ways that microcomputers can contribute to the quality of instruction and will also identify some of the problems that limit their usefulness. One of these problems--the scarcity of good programs or "the software problem"--is the focus of the second part of the paper.

I: POTENTIAL & LIMITATIONS OF MICROCOMPUTERS

As a student of elementary and secondary school curricula, I have witnessed the parade of reforms and innovations since Sputnik and have marched in many of them myself. Now, 20 years later, I would call myself a hardened veteran of educational revolutions. And yet I find myself excited by the educative potential of microcomputers--not just the dream machines of tomorrow but also today's Apples, Ataris, TRS-80s, and IBMs.

In an effort to avoid repeating my past mistakes, I have tried to assess carefully and objectively the potential and the limitations of today's microcomputers as educational aids. Briefly, I believe that today's microcomputers represent a substantial and exciting educational resource, but one with distinct limitations. And I believe that exploiting this resource will be a difficult and dangerous undertaking for schools.

The Potential of Micros Today

What can students learn or teachers teach using computers that they could not do, or could do only with difficulty, otherwise? This is the fundamental question. My experience--derived from three years of reading, thinking, and working with computers and computer-based education programs--leads me to identify

seven main ways that today's micro-computers can contribute to education. These are: 1) more active learning, 2) more varied sensory and conceptual modes, 3) less mental drudgery, 4) learning nearer the speed of thought, 5) learning better tailored to individuals, 6) more independent learning, and 7) better aids to abstraction. I will comment briefly on each of these capabilities.

1. More active learning. One of the most universal pleas of educational reformers from John Amos Comenius to John Holt is for more active learning. And one of the most consistent findings of educational research is that learning of all kinds is enhanced when learners can do something with what they are learning and see the results of what they have done. Computers can be programmed to call for repeated input from users and to respond immediately to that input. Thus students can attempt learning activities and can receive feedback on their attempts--a level of activity not possible while reading or listening to a teacher. This accords well with youngsters' inclinations to be busy, to get into things, to make an impression on the world around them.

Computers are interactive. A computer is not merely a medium for presenting materials to be learned nor simply a medium of expression for students. It is both, and more. A computer is also a device for processing a student's response and determining in a split second where to go from there. A computer can present a new stimulus that acknowledges a student's former response not only verbally but in content, appearance, and timing. Thus students clearly see that their input counts, that the program is responding to what they do.

2. More varied sensory and conceptual modes. Microcomputers as they come from the store can display letters and simple shapes in color; they can play single tones of varying pitch, duration, and loudness. They will accept input from typewriter keyboards, game paddles, joysticks, light pens, or digitized drawing pads. They can be connected electronically with any device that can be made to generate or respond to an electronic signal. For example, reasonably priced equipment is now available that can link microcomputers with videotape or videodisc players, electronic musical instruments, scientific instruments, physiological monitoring equipment, household appliances, and other computers. This makes a computer an enormously versatile teaching tool.

3. Learning with less mental drudgery. Mental drudgery is doing things that you already know how to do and that are not fun. It used to be widely thought that such drudgery aided moral development. Nowadays this view is out of fashion, but much drudgery still seems unavoidable in education--carrying out the intricate calculations involved in a budget or an analysis of scientific data, rewriting and retyping drafts of a paper, searching the library card catalog and then the shelves for the books one needs, and so on. Such drudgery eats away at the enthusiasm and energy we bring to otherwise interesting projects.

Utility programs such as a spreadsheet for numerical calculations, a word processor, and a data search and retrieval program can eliminate enough drudgery to make realistic problems manageable for youngsters. Once youngsters have learned to carry out the simple tasks such programs perform automatically, they can then concentrate on planning and strategic skills. Learning to use such utility programs should give youngsters a powerful set of tools for analysis--tools that require less time and effort and yield more penetrating and widely applicable results than all but the most talented, well trained, and disciplined of us can now achieve.

4. Learning nearer the speed of thought. When we encounter new information, whether live or recorded, that is presented at exactly the right pace for us, we find it exhilarating. Our attention does not wander, nor do we fall behind. We follow the train of thought seemingly without effort.

But we seldom experience this perfect match. For example, speech is usually much too slow. People can follow speech more easily when it is played back at a speed at least twice the normal rate. Yet, when we are struggling to absorb a new and strange idea, ordinary speech may be too fast.

Properly programmed, computers can match the pace and timing of a presentation to a learner's requirements at a given moment. The result is saving time; an average time saving of one-third is typically found in comparing computer-based education programs with conventional ones.[1] Moreover, satisfaction, pleasure, and confidence in learning increase. The computer makes one key goal of educational reformers--individualized rates of learning--routinely attainable.

5. Individually tailored learning. It is possible to program a computer to compose a lesson on the spot, tailored to the responses of just one student, using rules for selecting and combining preformed components according to the student's prior responses.

When learning difficulties are not remediable, as in the case of handicapped students, computers can circumvent individual limitations and concentrate instead on individual strengths. As the amount and sophistication of material to be learned grows, each of us will encounter our own limitations more often. We can all benefit from extending our performance with the aid of computers.

6. More independent learning. Anyone who has tried to learn a subject entirely without guidance from another person knows that the most serious challenges are verifying progress and sustaining motivation. Computers offer students new possibilities for independent verification of their own progress.

1. James A. Kulik, R.L. Bangert, and G.W. Williams, "Effects of Computer-Based Teaching on Secondary School Students," J. Ed. Psych., Feb. 1983, 19-26.

Programs can be written to monitor students' progress through lessons, to note errors and offer extra practice or more instruction where performance is weakest, and to offer help and advice whenever students ask for it. Instead of poring over a puzzling equation for hours, fruitlessly seeking the source of one's misunderstanding, a session with a good program would detect the problem and provide assistance quickly. For the highly motivated student, such help can make the difference between being able to learn independently and being forced to find a teacher--an option that is not always open.

7. Better aids to abstraction. The act of programming itself can be an aid to understanding abstraction. When we write a program, we represent complicated processes precisely and directly. Learning to write clear, complete instructions that enable a machine to carry out a task is every bit as difficult as (and perhaps more useful in an electronic age than) learning to do arithmetic computations. Writing even a simple program is a form of abstraction that is closely akin to action, and for that reason it may be more accessible to young people than the static symbolic forms we have traditionally struggled to teach them.

Computer graphics and computer simulation are also powerful new means of representing ideas and relationships in ways that permit us to act on them and to see the consequences. Computers can be programmed to create model worlds that operate according to a combination of strict rules and random processes and to give students complete control over them. These simulated worlds should be powerful aids to conceptual learning and thinking, enabling students to learn abstract relationships more easily than by merely reading about them.

The potential of computers in education is very great, but it is not unlimited. Let me now turn to some of the computer's limitations.

Limitations of Micros

The following seven problems seem to me to be the most severe limitations of today's microcomputers. 1) Microcomputers can supplement conventional education, but they cannot substitute for it. 2) Today's microcomputers are hard to use, and teachers prepared to use them are in short supply. 3) New products and systems are being created and marketed in such profusion, with such speed, and with so little standardization that systematic, long-term planning is nearly impossible. 4) Good programs are scarce because creating them for today's microcomputers is difficult, time-consuming, and expensive. 5) We are only beginning to understand how to use microcomputers in education; therefore, it is easy for a school or teacher to err, look foolish, or do harm. 6) Programs for teaching explicit, formal models can be created readily with known techniques, but it is much more difficult to use computers to teach subject matter that involves judgment, intuition, improvisation, and creativity. 7) Microcomputers will not solve (and may aggravate) several of the most serious current problems confronting education--notably equity, school finance, and divergent public expectations. Each of these limitations deserves some elaboration.

1. Supplement, not substitute. Independent study, home learning, and distance learning all have low rates of completion. Their chief problem is to sustain motivation and participation in the absence of a learning group, a teacher, and a social structure. The studies of computer-based education reported in the literature cover only uses of the computer as part of an ongoing class or school. Although computers have generally been successful in these studies,[2] the leap from this supplementary use to independent study via computer is a giant one.

Even though computers seem to ease the problems of distance learning, we are a long way from sustaining a mass educational system via computer. Indeed, we have not yet demonstrated that it can be done at all. At least for the next decade, computers will be strictly supplementary to conventional forms of education.

2. Microcomputers are hard to use. Our experience at the Microcomputer Institute for Educators at Stanford University during the past two summers indicates that highly motivated teachers can learn to use microcomputers for CAI, learn to program in BASIC, and learn to work with youngsters in computer literacy and computer programming courses in six weeks of full-time study, six hours a day, five days a week--a total of 180 hours. Perhaps this figure could be reduced somewhat by careful planning. But what remains would still require more hours of study than any single semester-long college course.

And such a course of study is only a beginning for an educator who wants to use computers professionally. Each computer has its own way of doing things, which must be learned anew regardless of prior experience. Each of the half dozen or so computer languages in widespread use requires further hours of study. Each new round of products is more "user-friendly," but it will be years before microcomputers are as easy to use as movie projectors-- machines that are already too complex for many teachers.

3. Rapid change and lack of standardization. When products are changing rapidly and each one has its own peculiarities, the return on the investment of time and energy in learning to use the system is low. As soon as you've mastered one system and become comfortable with it, another one becomes available that does 10 times as much and costs one-tenth as much. The only problem is that it will take you a year to learn the new system.

Schools do not have the resources to provide continuous inservice training for ever-changing computer systems, nor will teachers continue indefinitely to give up their spare time. In addition, software

2. Ibid., p. 123.

and accessories purchased for existing machines may not work with the next generation, and they will certainly be incompatible with competitors' machines.

Policymakers face the choice of investing in today's systems and forgoing the new ones, or, perhaps worse, they see that this same painful choice will continue to confront them indefinitely. This makes long-term planning nearly impossible and militates against the economies of scale that could come from coordinated programs of adoption and purchase. Perhaps standardization will eventually come to the computer industry. But I do not expect it within the next five years and probably not within the next decade.

4. Scarcity of good programs. Most teachers adapt curriculum materials and activities for their classes. Some create new activities, and a few of these write up their creations for publication in professional books or journals. Only occasionally is a teacher also a developer of commercially published curriculum materials. Developing software for computers is at least an order of magnitude more difficult than developing print materials that occupy a similar amount of instructional time. In order to develop software, a teacher must be able to program as well as to write. And a great many more instructional decisions must be made if the additional capabilities of the machine--timing, handling students' responses, feedback, and so on--are to be exploited.

Larger, more expensive microcomputer systems can make the software designer's job easier by providing such aids as utilities programs for graphics to make drawing on the screen easier; authoring languages to make programming easier; and standard subroutines to handle errors more efficiently. Within the next decade, even more sophisticated tools will become widely available, further simplifying the design of software. But educational software design will remain a highly skilled craft that few will master, and a shortage of good software will continue.

5. Nobody yet understands how to use computers well for education. The only thing that is absolutely clear is that the introduction of computers will require changes in the day-to-day operation of classrooms. Most of the time in today's classrooms is spent with teachers talking and children listening. (Two-thirds of all classroom time is spent this way, researchers say.)

The most extensive and ambitious evaluations of education programs that use computers have studied programs in which computers were used for only two or three hours per week for less than an academic year. Students generally worked by themselves on simple drill-and-practice routines or tutorials. We do not know what would be the consequences of extensive use of simulations for concept learning, of extensive use of computers for inquiry by small groups of youngsters, or of giving youngsters access to massive libraries of data. We will simply have to try different ways of using computers and carefully assess the results as we go.

6. Formalism favored over judgment. Computers handle rule-based procedures more quickly and accurately than any human being can, once the system of rules and procedures has been worked out. The field of artificial intelligence takes upon itself the task of discovering the extent to which those human capabilities we regard as intelligent can be represented in rule-governed procedural models and therefore programmed for computers to carry out. Perhaps at some point we will discover that everything we now regard as intelligent about human behavior can be programmed for computers. Consider the adjustment such a discovery would require in the self-image of our species. In the meantime, people can do many clever and important things far better than their ersatz kin--carry on conversations, make jokes, create art, invent art forms, build theories, infer intentions from behaviors, and, in general, handle novelty and ambiguity.

Using today's microcomputers, we can construct quite complex formal models, and these can be used to help teach students how to work with formal systems of knowledge. Mathematics lends itself to computer treatment, of course, but so do spelling, grammar, punctuation, those parts of science that treat formal theories and their consequences (e.g., the physics of motion or calculations involving chemical equilibrium), skills such as typing, masses of arbitrary facts (called data bases) that appear frequently in history and geography--in short, much of the standard curriculum of elementary and secondary education.

But not all. Literature, the arts, mathematical invention, scientific applications, those parts of science as yet unformalized, those parts of the social studies that require a broad or deep understanding of human interaction--in short, much of what has traditionally been considered the core of the humanities remains unformalized, not yet reducible to formal rules and procedures. Computers can be used to help teach these "soft" subjects, but their application is neither simple nor straightforward. And the difference between a student working with a computer and the same student working with a qualified teacher is much greater in these areas.

7. Computers will not solve current school problems. Computers will not bring racial balance to segregated schools or racial harmony to integrated ones. They will not redress inequities in funding between schools in rich and poor areas, nor will they overcome subtler inequities in the quality of education that stem from differences in race, ethnic group, gender, or socioeconomic class. Funding levels for education do not appear likely to rise because of microcomputers. Rival segments of the public will continue to hold contradictory expectations for schools and to struggle to impose their own views on the schools. Teachers will continue to be laid off, the teaching force will continue to age, and few talented young people will choose teaching as a career. Educators and educational leaders will still have to face these serious problems.

It is conceivable that a computer center could be run so as to harmonize the social divisions within a school and community. But unless special care is taken, computers are likely to be used mainly by middle-class male students for mathematics and science instruction. A talented, dedicated teacher or school administrator can counteract these tendencies and insure that all children have relatively equal access to computers. But this takes effort, planning, and cooperation, and nobody claims that microcomputers supply these commodities.

Since microcomputers play a supplementary role, they worsen budget problems. Even if the computers are donated to the schools, they must be maintained and repaired, people must be trained to use them, and software must be created or purchased. Thus computers still add to the cost of education.

The philosophical chasms that divide the various publics served by the schools also divide supporters and opponents of computers. Already the mark of progressive, open education can be seen on LOGO--and that of conservative educational philosophy on drill-and-practice programs. Already we can hear the clarion calls of educational sloganeers, such as, "Will the computer program the child or the child program the computer?"

As a new competitor for limited educational resources, computers are certain to arouse opposition. Computers threaten to disrupt established classroom routines and to put new pressures on teachers, so it would be surprising if some teachers did not oppose their introduction into schools. Certainly some defenders of humanist educational ideals and traditions are alarmed at the possibility that their notion of a good education will not survive the widespread use of computers.

Summing Up

Is it worth it, then? Are the limitations too severe and the advantages too slight? Readers must judge for themselves, but for my part the answer is a complex and qualified vote of confidence for computers in education.

For some communities, some schools, and some teachers, computers are not worth it--not now, not today's microcomputers. They can wait and see what happens, buying in later if things go well, when the hardware is more stabilized and the software better developed.

But it is very important, I think, for some communities, some schools, and some teachers to embrace computer-based education wholeheartedly, to strive to make it work for them. Success in using microcomputers for education will not solve the serious educational problems schools face, but failure will leave the schools even more poorly equipped to cope with them.

The microcomputer and its relatives, the other information technologies, are the new tools that happen to have been invented in our time. Learning to use them wisely and well is one of the major challenges we face. We have the opportunity to explore a new and very powerful medium of education and expression. How can we let that chance slip away?

II: THE SOFTWARE PROBLEM

When the time comes to use computers for something beyond computer programming and computer literacy, the software problem looms. There seems to be a great deal of software: simply wading through all the titles searching for what you want can be a day's work. But try finding some software to teach exactly what you need taught and you will, at least nine times out of ten, encounter the software problem.

Why is it so difficult to find good educational software? Is there anything a teacher or school administrator can do to alleviate the problem? Is it likely to get better in a few years? These are the questions I will consider in this section. In considering them I will need to begin with the more basic issue of how computers are used in schools.

Computers in Schools: Varied Patterns of Use

Computers can play a variety of roles in education, ranging from the most marginal of roles--as a supplementary optional activity for a few students, to the major role as a "teacher" of a course. Much of the excitement about computers in education is attached to the idea of the computer as a "teacher" in its own right--as a Socratic tutor, as a magnificent diagnostic device locating and remedying students' misunderstandings, or as the ultimate audio-visual device branching students through the Library of Congress on personalized learning paths. But the present reality is that computers play mainly a marginal role in schools except in computer programming and computer literacy classes and such vocational courses as typing, accounting, and electronics.

The software problem is relatively mild in these applications because teachers can use the software developed for more general purposes, such as operating systems, languages, word processors, accounting programs, and the like. But when we turn to important applications of computers to mainstream academic courses such as English, math, science, social studies and languages we find the software problem much more severe. We also find, and it is no coincidence, that use of computers as a major part of the teaching of these academic subjects is quite rare today.

That the software problem is most severe in mainline academic subjects is an important clue to some of the origins of the problem. In those educational applications where computers are now being widely used--computer literacy, programming, vocational applications--generally only a single piece of software is needed, an operating system, a language, or a word processor. In the teaching of an academic subject like algebra, however, perhaps as many as a dozen or more substantial programs will be needed if the computer is to be useful over the entire course. If we multiply the number of subjects taught (perhaps an average of 8 per year) by the number of grade levels (12) by the number of programs needed per course (say 10) we see that nearly 1,000 programs are needed simply to cover the public school academic curriculum with only one program per topic.

So, even though software seems to be flooding the market--our files at Stanford include over 100 catalogs of educational software with more than 3,000 titles--coverage of the software needs in academic subjects remains spotty. When you consider that most of the items in the catalogs are concentrated in a few subjects and topics (elementary math drill and practice, spelling, and computer literacy), it is easy to appreciate the enormous variety of the demand for software. This variety means that the market for any single piece of software filling only one of these thousands of niches will be smaller than the market for more generally useable programs. Until the number of computer-using teachers in the various subjects increases substantially, the market for software in those subjects is too limited to justify an investment in producing it. And, so long as the selection of software is limited, many schools and teachers will be reluctant to use computers in their teaching. This is the familiar Catch 22 situation that confronts any innovation, but the fragmentation of the market into so many niches makes the problem more severe in the case of substantial applications of computers to mainline academic subjects.

Good Education Software

Computers have been called "chameleons in the classroom" because they can be used in so many different ways. Computers used for drill and practice with individual students seem so different from the machines used as an "electronic blackboard" to present animated geometry diagrams or from the machines used by a small group of students in a simulation game. The differences are produced by the software: the computer may well be the same machine in all these applications.

The qualities that make a computer work well for one of these educational uses may not necessarily be desirable in the others. And this is another source of the software problem: varied criteria. A piece of software that does a good job of teaching arithmetic facts through drill and practice will not satisfy educators who want programs that develop understanding of number concepts. A program that entertains and motivates students with color graphics and animation will please those whose educational philosophy is child-centered and displease those whose philosophy is more subject-centered. Such diversity of criteria further reduces the likelihood of finding software that will be generally regarded as good and increases the number of niches in an already fragmented market.

Finally, we must recognize how high are the standards typically used to judge educational software. Few dispute that computer programs can teach number facts, but we also know that traditional methods such as flashcards, can do the same job and much more cheaply. Computers, being more costly, must accomplish more than traditional methods if their use is to be justified economically. By extension of this line of reasoning, ways must be found to use computers to teach the most difficult concepts and skills, those which substantial numbers of children now fail to learn using traditional methods. To develop programs that achieve these high standards is not an easy task. We certainly cannot expect that anyone should be able simply to sit down and write such programs. They require thorough analysis, deep thought, and inspired design.

Current Dimensions of the Software Problem

The fundamental problem is a shortage of educational software that can be used as a major part of the teaching of academic subjects in elementary and secondary schools. The number and variety of programs needed to alleviate this shortage is large, but, as the saying goes, "you ain't seen nothin' yet." We have yet to consider several other aspects of the problem that make it larger and more severe than it seems so far. The aspects that follow are presented in no particular order.

1. Development time and cost. The best estimates of the time required to design and code a computer program range from 100 to 300 hours per hour of running time. This does not include the time needed to think up the program ideas. This translates into a development cost for a program that students might use for one hour of between $2,000 and $100,000, depending on its sophistication and complexity. By contrast, to produce text material to occupy a student for an hour is a matter of a few hundred dollars at most. And remember that the market for the software is limited by the number of machines available and the large number of small niches in the market, much greater limitations than apply to text materials.

2. Machine incompatibility. A new form of Murphy's law: The program you want is only available for a machine you don't have.

3. Software piracy. Software manufacturers are reluctant to invest in the development of products that will be copied at no charge by the customer. If one can sell only one copy of a program per school, the price necessary to recover the investment must be large, between $300 and $500 per copy. This, obviously, makes it prohibitively expensive for a school to buy enough copies to supply one for each computer and therefore ensures either that the software will only be used as a supplement or that it will be illegally copied.

4. Locating and reviewing software. Even when good software exists, finding it and verifying that it is good are nontrivial problems. Indexes are beginning to appear that list software by subject, grade, and other useful properties, but at this moment coverage of such indexes is spotty. A number of journals publish reviews of software, but finding a review of the program you have in mind remains difficult. What we need are specialized publications that review programs in a small area with particular reference to their usefulness in the classroom. Again, we are confronted with the problem of a plethora of small niches which make it uneconomic to provide reviews to such a small audience.

5. Competition for the home market, rather than the school. The number of installed machines in homes far exceeds the number in schools, and individuals buy a total dollar volume of software several times greater than schools. Software manufacturers can therefore sell to a larger market by producing for the home. And most of them do. This means that software is designed primarily for conditions in the home--one student per computer, unsupervised use, episodic use with little extended continuity in the development of skills and ideas.

6. Problems in integrating software into the classroom's other activities. Even a well-designed piece of software will not fit exactly into a given teacher's plans. Adjustments must be made to accommodate it. If the software is not modifiable, then all the adjustments must be made elsewhere, and there are limits to a teacher's willingness to tailor everything else to one program. And when a teacher uses several programs in the course of a year, each of which requires a different set of adjustments, the problem may become insurmountable. Examples include the spelling program whose words do not match the teacher's goals, the math program which introduces skills in a different sequence from the school's curriculum, and the science program which uses a different notation from that in the textbook.

All these difficulties translate into a higher cost to provide the software needed. The cost of equipping a single course in one school with enough software to be used one hour per week for 30 weeks in a school year, assuming appropriate programs were available at today's typical price of $50 per diskette, and an optimistic 'playing time' of three hours per diskette, and one diskette for each three students in a thirty-student class, comes to $5,000. This figure is too expensive by a factor of ten. So long as costs are this high, the market in schools will be thin.

What To Do?

What can be done today to overcome the software problem? The software problem manifests itself as an economic problem, even though not all of its causes are economic. The home market for educational software programs will likely continue to be bigger and richer than the school market, and therefore software companies will continue to produce for that market. Eventually, competition for that market will make the smaller niches in the school market relatively more attractive, and we will then see more production of software specifically for schools. In the meantime, however, the home market is far from saturated, so the present situation is likely to continue for some time.

Only large-scale actions would change this economic situation substantially. If the government and private foundations could be persuaded to finance dozens of software projects in education, that would make a dent in the problem. If districts formed consortia and invested their own funds in the development of software, that would have a significant impact. Million dollar contracts between software development houses and school districts to develop software collaboratively and share royalties would have an effect. And the simple growth of a school market for software will, in itself, stimulate more and better software. Progress in this fundamental aspect of the problem requires growth of investment or expenditures or both. If this growth fails to come or comes slowly, the software problem will not improve, regardless of anybody's good intentions or hard work.

In the present thin market, buyers' decisions have an immediate and powerful shaping effect on producers. Those products that sell will be widely imitated, while those that do not will rapidly disappear from the catalogs. Schools can influence the future direction of growth in software by being discriminating buyers. Buy only programs that give you the most for your money. One measure of the value of a piece of software is the number of student-hours of use per dollar of cost. This figure ought to be computed in reviews of software prior to every purchase. Another important quantitative indicator is the extent of your curriculum covered by a program. One that is useful in only 1% of a year's classes is less valuable than one useful in 10% of classes.

What actions can an individual school or district take to cope with the software problem? One thing that should be done is old-fashioned curriculum development. Scope and sequence charts are needed showing just where what types of computer programs can be used and teacher's guides showing how they can be integrated with the other ingredients of a good course.

The problem can be eased by extensive use of tool-type programs and modifiable programs. The Music Construction Set is a program that transforms a computer into a composer's typewriter. A staff and various symbols are displayed on the screen and these can be moved around with keyboard commands or a light pen to compose music which can then be played by the computer with the press of a button. Such a program can be used throughout the year in a music class. A spelling program which permits teachers to enter their own words is much more valuable than one with a fixed word list. Insist on programs that can be used as tools by teachers and students, programs that can be tailored to suit your curriculum.

One type of educational software that is little known in this country but widely discussed in Europe and Japan can be used to great effect in academic classrooms. These programs are called "electronic blackboard" programs. One such program, Quadrilaterals, is published by Readers' Digest. A teacher uses this program on a single computer at the front of the classroom, with a screen large enough for all students to see. Using game paddles, the teacher is free to walk around the room while controlling the display. The teacher can choose to have text displayed or only diagrams. Questions can be posed for class discussion and then the animation powers of the program used to show the answer on the diagrams. The program is used very much like a chalkboard by the teacher, so that no extensive inservice is necessary to prepare teachers to use it. Such programs can be extremely cost-effective ways to use computers in the teaching of academic subjects.

Teachers can create their own educational software. To do this 'from scratch' in Basic or assembly language is a difficult and time-consuming activity that cannot be expected from teachers working full time. But, using an authoring system such as Pilot, teachers can develop lessons in only slightly more time than it takes to develop ditto masters or overhead projection sheets. However, to develop software that is truly interactive and that accomplishes things conventional methods cannot remains a high art, difficult and time-consuming. It might be reasonable to expect a talented, dedicated teacher working with an authoring system after school, weekends, and holidays, to produce two or three hours' worth of such programs in the course of a year, but not more. Unless your school has an unusual concentration of these rare birds, it is unwise to rely upon teacher-made computer software for a major part of your courseware. If you are determined to rely on teacher-made software, you might consider contracting with the most able teacher-developers to spend a substantial portion of their time for a year or so developing software. You might even consider entering into a consortium with neighboring districts to pool the talents of your teacher-developers.

It is possible to lease or purchase a complete set of integrated software designed for school use. Computer Curriculum Corporation, for example, has complete computer-administered courses in most of the subjects of elementary and secondary schools. Many publishers of basal texts for the elementary school also offer computer software designed to accompany and enhance their text materials. Control Data Corporation's PLATO system offers a good selection of software for most school subjects. These integrated software systems are expensive, and they are not tailorable to an individual school's or teacher's needs, but they may be more cost-effective than assembling your own software from catalogs or developing teacher-made materials in many instances.

Over a somewhat longer period and on a larger scale, you might contemplate entering into collaborative arrangements with software developers to work on the most pressing software development needs. Consortia of districts contracting with private developers and involving local teachers in the development process can be a powerful development strategy. Collaboration is also possible with professional associations. Such initiatives put your school into the software development business and this might be difficult to arrange with your board, but they give you more control over the software than you get any other way.

In summary, the software problem is serious and can be traced to some fundamental economic causes that are not easily overcome, but there are constructive ways to cope with the problem if you are willing and able to invest the money, time, energy, and initiative. The problem will not go away in the foreseeable future, in any event, and it will only get substantially better as more organizations invest more in developing good educational software.

Selected References on Computers and Education

GENERAL

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

Coburn, Peter, et al. (eds.) Practical Guide to Computers in Education. Menlo Park, CA: Addison-Wesley, 1982.

Bitter, Gary and Ruth A. Camuse. Using a Microcomputer in the Classroom. Reston, VA: Reston Publishing Company, 1984.

Walker, Decker F. "Reflections on the Potential and Limitations of Computers for Education." Phi Delta Kappan, September, 1983.

CAI

Chambers, Jack A. and Jerry Sprecher. Computer-Assisted Instruction: Its Use in the Classroom. Englewood Cliffs, NJ: Prentice-Hall, 1983.

EVALUATING EDUCATIONAL SOFTWARE

Lathrop, Ann and Bobby Goodson. Courseware in the Classroom: Selecting, Organizing, and Using Educational Software. Menlo Park, CA: Addison-Wesley, 1983.

DEVELOPING EDUCATIONAL SOFTWARE

Walker, Decker F. and Robert D. Hess. Instructional Software: Principles and Perspectives for Design and Use. Belmont, CA: Wadsworth Publishing Company, 1984.

Bork, Alfred. Learning with Computers. Bedford, MA: Digital Press, 1981.

The Improvement Support Program at Far West Laboratory provides a variety of educational services to educators in Northern California, Utah, and Nevada including research information, technical assistance, and staff development.

Lynn Jenks	Principal Investigator
Carol Murphy	Information Coordinator
Stanley Chow	Technical Assistance Coordinator
Kendra Bonnett	Technology Coordinator
Tom Ross	Writer
Ellen Hui	Program Assistant

A limited number of additional copies of this Brief is available from the Improvement Support Program.

IMPROVEMENT SUPPORT PROGRAM
Far West Laboratory for Educational Research and Development
1855 Folsom Street
San Francisco, CA 94103

Non-Profit Org.
U.S. Postage
PAID
San Francisco, CA
Permit No. 10660