

Computers in Education: Why, When, How

Excerpts from *Challenging the Applications: An Alternative View on Why, When, and How Computers Should be Used in Education*, where more details and a comprehensive bibliography may be found.

by Valdemar W. Setzer and Lowell Monke

Introduction

Since the 1970s, computers have been inserted into schools with much fanfare and great expense. Almost all of the attention to computers has revolved around the question of how to put them to use as tools for education—with the superficial response being: every way possible. Subsequent assessments have focused primarily on their effectiveness as an aid to learning and on how well students are prepared for their use beyond school. Through all of this, almost no attention has been paid to the more fundamental issue of the effects that the very use of computers has on young people. A determination of the proper role of computers in education must begin with an understanding of the computer's effects on the user. This basic concern will serve as the point of departure, eventually broadening to a critique of current uses and recommendations for governing the use of computers in education.

The Characteristics of the Computer

Computers are very special types of machines, completely different from others. Whereas others do physical work, computers do not: They process *data*, which consist of specific kinds of thoughts introduced into them. One should not confuse *data* with *information*. Information has meaning, "semantics." Data are just symbols. For instance, the number 2000 is a sequence of 4 symbols, which has no meaning in itself—it is just data. It may, however, be associated by the computer user to a salary, and thus acquire a significance—completely "unknown" to the machine. At most, inside the computer one may associate the number 2000 to the string of letters "salary" through physical contiguity or through "pointers." Through this *structural* (which may also be called syntactic) association one may couple these two pieces of data, but what both "mean," that is, their relation to the real world, cannot be inserted into the machine. So, computers work with an extremely restricted class of our thoughts, thoughts that do not have the same meaning to the machine that they represent to us (indeed, to the machine they have no meaning at all).

Computer programs are also thoughts that the programmer inserts into the machines, which process the thoughts that are represented by data. In contrast, a power lathe acts directly on the physical world, transforming some matter. A telescope transforms the light that traverses it. A hydroelectric plant transforms water's potential energy into electric power. A car is used to

transport matter (people). A battery stores electric energy. Thus, one may say that other machines transform, transport, and store matter or energy, that is, physical elements. Computers, on the other hand, transform, transport, and store data, which are not physically consistent because they represent some special types of our thoughts. (It is not possible to grab human thoughts, measure them, observe them with our eyes or instruments.)

This symbolic manipulation of data characterizes computers as “abstract machines,” as “mathematical machines.” In fact, it is possible to describe with the formalism of mathematics all data processing done by a computer. It is also possible to do any data processing mentally or by using pencil and paper, as long as the computer is not controlling other machines. Programming a computer corresponds to elaborating purely mathematical thoughts. It is a process analogous to theorem proving. Although it is not so obvious, this is also the case when one uses any software, as for example a word processor, where one has to exercise exactly the same type of reasoning used in algebraic mathematics. To solve an algebraic equation one must work formally, logically, step-by-step, through a set of operations *predefined by the algebraic system*. Exactly the same situation occurs when undertaking a complex formatting activity with a word processor, relating cells in a spreadsheet, creating reports in a database, drawing with a graphics program or issuing commands to an Internet browser. One should keep in mind that in each case, including the mathematical one, the system restricts the options available for use to those relatively few defined operations that make up the system. It is not possible to achieve legitimate results by stepping outside that system. Thus, the mathematician, the computer programmer, and the computer user all find themselves constrained to a logical, formal, and extremely narrowed thinking environment.

The shrunken thinking environment of the computer, whether exhibited through programming or the menu (or even iconic) commands used in word processors, spreadsheets, and so on, are examples of what one calls *formal language*—a language with a strict syntax, which may be fully described in mathematical terms. Besides being formal and impoverished, such languages have another very important difference from *natural languages* such as Portuguese and English: They are unambiguous. This means that each instruction or command interpreted by the computer produces the execution of exactly one specific function related to the data. Let us emphasize: This function may be mathematically—that that is, exactly—described. On the other hand, the meaning implied by a phrase in some natural language may be ambiguous and cannot be mathematically described, particularly when it refers to the real world. For instance, how should one interpret the following phrase: “The vase fell on the table and it broke.” What broke? The vase or the table? The computer cannot deal with such types of ambiguities. Each command, such as letting the vase fall, has to have exactly one “meaning,” that is, a function it performs on specific data and the unique result it produces. The computer is a strict “syntax machine;” every program and piece of data has to be described internally in a purely structural way, using some representation through formal, mathematical symbols having one single, mathematical interpretation. Otherwise, one could get different results every time one processes some program using the same input data; the computer

would lose exactly the essential deterministic characteristic that makes it valuable to us. There is no possibility of representing our “semantics”—what does “breaking” mean? Cutting into pieces? How many? And even less our “pragmatics”—if one knew that it was a sturdy wooden table, probably the vase broke.

Moreover, the type of thinking necessary to program a computer or to use any software is the same as the thinking used when doing symbolic logic. Programming and command languages demand that the student reduce decision-making to a process of making logical choices by stringing together exact and unambiguous expressions. This requires that the student operate in a cognitive straightjacket.

Because computers are mathematical machines, forcing a purely abstract and mathematical type of thinking as well as a symbolic formal language, we think that they should not be used by children in any form before approximately age 15, or high school, for it is at this period that the young person reaches the intellectual maturity so that thinking forced by the machine is not detrimental to development.

When students are introduced to computers in high school, it is imperative that they be helped to understand how computers work, how they work on us, and how we may properly work with them. We believe that a phenomenological study of the computer hardware can, and should, be undertaken by students sometime between the ages of 15 and 16. Most students are fully capable of drawing conclusions from observing the electrical and mechanical operations of the hardware of the computer by that age. They would be learning about how a machine they see almost everyday works, starting from its simplest, most concrete activities.

Due to the degree of abstraction and self-control required for using computer software, we would recommend around age 17, during 11th and 12th grades, as the ideal age to cover software, from basic operations to general applications, as well as a general investigation of the effects of computers and high technology on society and the individual. Software would only be learned after a basic understanding of computer logic and the principles behind basic programming. The rationale behind this sequence of learning, as in any other subject, is that students would be learning hardware and software in a bottom-up fashion. Furthermore, they would be following the historical evolution of computers and their use: databases were in common use only by the end of the 1970s, word processors, spreadsheets and graphics became popular only in the 1980s, with telecommunication taking hold in the 1990s.

We recognize that many people, even those who agree with our earlier discussion, will find this timetable disagreeably slow. We are not unaware that young people are able to do very sophisticated work on the computer at a much earlier age. But our argument throughout has been that computer exposure should not be based on capability but on developmental appropriateness.

The “capable child” is a trap that the current technological society lays at every turn. We must always be as alert to what will be lost by the introduction of a new way of thinking into a child's life as we are about what might be gained. Before the last years of high school, few young people have the emotional maturity and, perhaps more crucially, the critical self-awareness to help protect them from sliding into the cognitive muscle-boundness that is promoted by the use of computers.

Dangers of early Computer Use

Children are fascinated by computers, as they are fascinated by any complex device that responds to their attempts to manipulate it. Being the master of a complex machine and exploring one's capacity to dominate it, may explain why computers exercise such tremendous fascination. But there is also a danger that this fascination will lead to obsession. According to our experience, only at around age 16 to 17 do young people have enough self-control to avoid falling into the obsessive programmer's or user's state.

Even when obsession is not an issue, we must be careful not to be misled by the purposefulness demonstrated by children using computers for learning. The fascination they display is generally not with the learning experience or with the subject matter at hand, but with the operation of the computer itself. While teaching computer based Geometry classes, Monke observed that his students were generally quite successful at discovering geometric properties using a mathematical drawing program, but once they walked away from the computer they often could only recall what they drew, not the mathematical principles they found. This is a problem with all complex technology. It has the tendency to draw attention away from its intended use, inward, toward itself.

Although the consequences are certainly not so dire, the problem of mixed purposes also holds true for children using computers in school today. Regardless of the subject matter, the primary learning experience, and the one on which the child's attention is focused, is how to manipulate the computer. The child often doesn't really care much, if at all, about learning the material presented. The main concern is to spend time using the computer. The teacher's goal of getting the student to arrive at a knowledge of, say, history is never internalized as a goal by the student, for whom the destination is secondary to the trip. This is one of the reasons studies of computer-aided learning have shown such decidedly unspectacular success, given the incredible expenditure of funds, resources, and effort put into many of the projects. When the content is secondary to the medium, we can be assured that the content will not stay in the mind for long.

Even so, educators look to computers as the one strategy that can at least engage the student in some form of learning. In these instances, the computer is viewed as an artificial sweetener, used to make what has become the bitter medicine of learning palatable to children raised on the empty calories of TV. We believe that using the computer as an artificial sweetener of learning is

pedagogically dishonest. It introduces a harmful additive to the educational diet that only temporarily covers up the sour taste that too many children have toward learning. Using computers in this manner is hardly the way to instill in children a love for life-long learning.

There has been one benefit from the fascination shown by children for computers in education. Perhaps the computer's greatest service to education has been to make obvious to everyone what critics have been complaining about for decades: poorly trained, unenthusiastic teachers, using poor methods based on faulty philosophical foundations to teach irrelevant material makes for boring education. The computer has sounded the alarm for all to hear, but that does not mean that in it lies the solution to the problem. It is, in fact, a sad irony that the computer can attract many students' attention better than a teacher can; for the human personality is potentially infinitely richer, and has an individuality and sensitivity which are absent from any apparatus. Turning to the computer for help because it can process abstractions in swift, attractive ways, shows infinite patience, does exactly what it is told to do, and does not assign bad grades, is simply a matter of finding a more seductive means of teaching the wrong way. Thus the computer takes us further down the road toward remaking education into a mechanical, solely intellectual activity. As education gets redefined in this much more limited concept of information processing, the teacher and all his or her materials cannot hope to compare favorably with the ultimate multimedia device. Thus, the computer becomes not only the tool of education but its model.

Many educators tout the value of *unrestricted software*, because it facilitates problem-solving through trial and error. (By unrestricted software we mean those programs that give the user a virtually unlimited number of choices through the combination of commands.) There are certainly some situations in which trial and error is appropriate, but in most typical cases where trial and error is employed in day-to-day work on a computer, it reflects a lack of mental discipline.

Because software commands are, in fact, thoughts, and there are no physical constraints to them, one may construct a text using a word processor in a completely undisciplined manner. Developing the ability to structure a cogent pattern of thought before trying to express it is one of the most painstaking and unceasing tasks of good schooling. Handwriting, and even typewriter typing, demand—and therefore, encourage the development of—well-structured, disciplined thinking by allowing for only few and small corrections. This is not the case with word processors: They permit the user to type a text without paying much attention to it, because all sorts of changes and corrections, such as deletions, pasting, formatting, and so on, can be made later. There is no need to pay attention to spelling, because the automatic speller indicates errors and suggests possible corrections; reasonable grammar checkers are also already in use.

We are not advocating that everyone return to using the typewriter. But we find it disturbing that many children are being taught such things as word-processing and LOGO programming at the very time that they need to be developing disciplined thinking habits. By relying on the computer to

overcome disorganized thinking we may see perfectly spelled, neat documents produced, but what we do not see is what happens, or rather does not happen, in the child's mind. Just as a child's physical development is stunted when muscles are not exercised, the development of disciplined thinking is stunted when the computer relieves children of the responsibility for planning and organizing their thoughts before expressing them. We should always keep in mind that tools that are designed to aid the mature mind may hinder the maturation of the developing mind.

This leads to another problem as education becomes computerized. We have seen that in using the computer the child is forced to think in a way that is appropriate only for adults. Thus, one could say that computers contribute to the compression of childhood. It is similar to the condition David Elkind (1981) described in *The Hurried Child*. Hurrying children may allow them to display a certain intellectual prowess, but creates emotional and psychological stress that harms their development and shrinks their childhood. Postman (1988) believes that this is not just an individual problem but a cultural ill that may be fatal to childhood itself. He argued that easy access to adult information without having adult sensibilities leads to behaviors and attitudes that are rapidly eroding the very concept of childhood.

Education has always been highly contextual: before buying a book or a toy for her child, a parent usually examines it to see if it is appropriate to her child's level of development and if it agrees with her pedagogical concepts of what is proper for children at that age. A teacher always teaches taking into account what she taught the day before, the past week, months and even years, as well as the particular condition her class is in at that moment. Nothing of this sort happens with computers. In particular, the Internet is absolutely un-contextual and requires a high degree of maturation for the user to choose what is adequate for her learning process.

Clearly, the computer has only exacerbated these problems. It blows wide open the doors to the adult world of information. Even worse, it allows children actually to participate in that world on an equal footing. While some researchers hail displays of adult-like, rational thinking, we fear that children and young people will be crippled by the too early suppression of their natural, intuitive, open, holistic way of relating to the world. We fear that by employing the computer to move away from their natural way of understanding too soon, "machine thinking" will come to dominate the way children regard nature, their fellow human beings, and life itself. By "machine thinking" we understand the type of thinking necessary to create thoughts that may be introduced into a computer, in the form of software commands or instructions in a programming language, so that the computer may correctly interpret them. This is a threat that is unprecedented, for there has never been such a strong metaphor as the computer for the image of humans as machines and machines being able to behave as humans. We think that machine thinking, especially when instilled in young people at an early age, leads to this mentality, with negative consequences.

Computer Education as Preparation for Work

We believe that at the appropriate time, as befits its centrality in our culture, the study of technology must assume an important role in the young person's education. But that is not the same as advocating early instruction in how to use computers or incorporating it directly into the learning experience. The parental concern that the child will be at a disadvantage if not started on computers early is a fear—exploited shamelessly by software and hardware computer companies and some computer-based commercial “educational” services—that simply does not have basis in fact. Let's briefly look at the issue from two perspectives.

Setzer taught computer science at the university level. At his institute, he recognized that those students who had lots of contacts with computers before getting to the university often had considerable difficulties: they tended to have very little patience for learning the skills that constitute real computer science - data structures, computing theory, program development and documentation, and so forth - because they were used to using sophisticated software. Once they got into computer science they found that their homework had nothing to do with, for example, drawing spectacular figures on the screen; that it was a serious and laborious activity, requiring great effort and concentration, and that it was nothing so easy as the playing they had done with computer software. Our experience is that early contact with computers gives a totally wrong impression of what computing and software development is, disturbing a serious study in this direction at college. It is often necessary to unteach bad habits before a really serious learning of computer science can take place. For these specialists, it is better that they build up their creativity, their disciplined thinking skills, a well-rounded grasp of the physical world, and a strong, incorruptible sense of humanity at the secondary level, than invest thousands of hours of computer time having nothing to do with the hard mathematics and labor involved in real computer science.

But most young people will never become computer scientists. What of those who need computer skills in order to use them in the workplace, i.e. the vast majority of students who pass through our schools? For nearly two decades Monke taught computer applications to this broad spectrum of students. He found that in one semester, working 45 minutes per day, a typical young person can easily master the standard features of a word processor, spreadsheet, and database, with time left over to experiment with such things as telecommunication, drawing/painting programs, or desktop publishing.

Given the ease of use that is currently being built into software, even this amount of instruction is more than sufficient to develop the basic computer knowledge needed to enter the general workplace, the technical school, or the university. We think that when it comes to software, schools should concentrate on teaching concepts and showing what may be done with computers without entering into highly specialized training activities. It has been our experience that any work beyond these fundamental aspects of computer education becomes so job specific that it cannot be

justified at the secondary level. Greater details should be left for self-learning or should be provided by the enterprise or college. This will be facilitated by the constant evolution of good tutorial programs, as well as good help features incorporated into the software.

Introducing Young People to Computers

In the following section we propose a full curriculum for the introduction of computers in the last three years of high school. We propose the installation of “Technology Laboratories” in high schools, where students learn how machines work, including the telephone, steam engine, combustion engine, electric motors, radio, TV and, obviously, computers. A strictly practical approach should be followed in these classes, leaving theories for the physics and chemistry classes. The fundamentals of computer—and calculator—hardware may be taught by stressing the phenomenological aspects, with very little theory. The important properties of digital circuits and components may be deduced from simple experiments performed by the students; that is why we are proposing these classes for younger ages. We believe that this lab experience should be part of the core curriculum for all students. Without at least a fundamental knowledge of how a technology works a person has no chance to assert control over it and will be prone to the apathy discussed earlier. As the inner workings of everyday devices become more and more opaque it is essential that the educational experience shed some light on how these things operate.

10th Grade: In the Technology Lab, introduce DC electric circuits with batteries and resistors, LEDs, magnets and relays. Logical gates with relays; simple applications, like a circuit with different switches for the same purpose (e.g., various buttons to move the same electric window of a car), "optimized" traffic lights, and so on. One step in demystifying computers is understanding some of the fundamental differences in natural language and the language we use in talking about, and working with, computers.

11th Grade: In mathematics, after progressions and logarithms, introduce binary and decimal numerical bases, base conversion and binary addition and multiplication. In the Technology Lab, redefine logical gates; introduce “flip-flop” with a relay and show how it may be used for storing binary digits; introduce diodes and transistors in a purely practical, functional manner, redo adders and storing devices with them. After this, introduce in the Computer Lab the computer's basic components and Machine Language; execute and modify simple Machine Language programs simulated in a computer.

12th Grade: Computer Lab: introduce programming languages (such as BASIC, Pascal or LOGO), not in an effort to develop skills in programming but in order to understand how they work; teach the fundamentals of word processors, spreadsheets, graphics and database systems, explaining internal structures when possible; notions of computer networks; practice with Internet browsers, chat systems, usenets, electronic mail, remote file transference and remote access to general

databases. In mathematics, introduce the notion of algorithms, stressing the necessary quantification of data and programs introduced into computers. We believe that it is essential that in social studies or philosophy classes a substantial study of the individual and social impacts of computers take place at this time due to the importance they have acquired.

Conclusion

Computers have penetrated every human activity. Problems caused by them are often neither direct nor visible. Computers not only aid (and replace) thinking but can shape it. For all these reasons we have to be extremely careful in using them in education. We have to educate for their use with much more care than other machines.

We have stated a case for reserving the gradual introduction to computers for the last years of high school. Obviously, we do not see computers as the saviors of education. School systems are doing a very poor job of preparing young people to lead meaningful lives, but this universal problem is a human one, not a technological one. The school of the future need not be a more technological school, but it must be a more humane school. Rather than discarding ancient traditions out of arrogant disregard for the past, the school of the future must honor those traditions as the foundation of humane education, building on them through our deep observations and understanding of the present human condition to reorganize and improve old structures and institutions. This is not a radical proposal. Waldorf Education has been doing this since 1919. It is common practice in Waldorf Schools not to introduce computers until high school.

We think the school of the future should have human teachers and classrooms, but teachers will have to fight courageously to resist the pressures—by bureaucrats, by commercial interests, by psychologists and by politicians—to turn them into technicians, information repositories, transmitters and facilitators, or that horrible new expression “liveware.” They will have to relate to their students as human beings in development, and not as storing and sorting machines, as real individuals, and not as collective abstractions. Whereas computers handle all their users exactly in the same impersonal, cold manner—as machines—only a human being can respond to a child out of a deep personal knowledge and intuition of the child’s needs, aspirations and moods. Only a human teacher can recognize a correct knowledge in an incorrectly formulated answer. Students need understanding, compassion, love, and sacrifice from their teachers far more than they need access to billions of bits of information. They urgently need to admire their teachers as individuals with knowledge, life experience, and insight, (i.e., wisdom), for the problems of children and youth. Students need more than mere trainers of skills—they need teachers who can help them develop and appreciate those noble qualities that have always formed the core of what is best about being human—qualities such as social responsibility and sensitivity, compassion, courage, love, sacrifice, honor, and justice. This cannot happen as long as schools regard teaching as a science, technique, industry or commerce, instead of an art.

We believe that early computer use and an emphasis on computer-like thinking, is leading children's development to be dominated by the rigid, logical, algorithmic thinking, bereft of moral, ethical or spiritual content, that is characteristic of computer interaction. This accelerated, but isolated intellectual development brings a child's mental abilities to an adult level long before the emotional, psychological, spiritual and moral sensibilities have grown strong enough to restrain it and give it humane direction.

What will be the consequences of this disrespect toward children's nature? We fear that as these children are evaluated and encouraged to see themselves more and more according to these limited cognitive qualities, their respect for themselves and the human race will be further eroded. For humans cannot compete with the computers in this one narrow range of mental activity. This is, perhaps, the most frightening consequence of the advent of using computers in education: the inducement to admire, venerate, depend upon, and finally raise above ourselves the machine; to view them as superior to ourselves and to view ourselves as merely imperfect machines. A future based on such a world view is terrifying, for ethics, morality, justice, and mercy are all irrelevant to the machine. As needed they may all be “logically” sacrificed in the name of the gods of technology: efficiency and productivity.

Our hope is that the introduction of computers only after a childhood environment steeped in love, beauty, and respect for children's natural, holistic growth may make it possible for them to put these machines in their proper place. We have tried to outline a framework for handling that introduction with hopes that others will refine it, adapt it, and make it a viable program for their schools. We recognize that it will take courage to withstand the pressures against it. Perhaps the most important thing is to try. Right now more than anything else we need more voices challenging the trend toward technological dominance of education. We hope that the ideas in this essay can provide support and encouragement for that endeavor.

References

Bowers, C.A. (1988). *The Cultural Dimensions of Educational Computing - Understanding the Non-neutrality of Technology*. New York: Teachers College Press.

Elkind, D. (1981). *The Hurried Child - Growing Up Too Fast Too Soon*. Reading: Addison-Wesley.

Finser, T. (1994). *School as a Journey - The Eight-year Odyssey of a Waldorf teacher and His Class*. Hudson, New York: Anthroposophic Press.

Gabert, E. & Niederhaueser, H.R. (Eds.). (1962). *Konferenzen Rudolf Steiner mit den Lehrern der Freien Waldorfschule in Stuttgart 1919-1924, Vol. 1*. Private edition: Stuttgart, (translation unknown).

Kline, M. (1973). *Why Johnny Can't Add - the Failure of New Math*. New York: St. Martin's.

Papert, S. (1993). *The Children's Machine - Rethinking School in the Age of the Computer*. New York: Basic Books.

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

Postman, N. (1988). "The Disappearance of Childhood." In *Conscientious Objections* (pp. 147-161). New York: A.A. Knopf Inc.

Postman, N. (1993). *Technopoly - The Surrender of Culture to Technology*. New York: Vintage Books.

Roszak, T. (1986). *The Cult of Information - A Neo-Luddite Treatise on High-Tech, Artificial Intelligence, and the True Art of Thinking*. Berkeley: University of California Press.

Setzer, V.W. (1989). *Computers in Education*. Edinburgh: Floris Books.

Setzer, V.W. (1992). *Computer in der Schule? Thesen und Argumente*. Verlag Freies Geistesleben, Stuttgart .

Setzer, V.W. (1993). *Tietokoneet ja Kouluikaeiset? Vaeitteitae ja Perusteluja*. Tampere: H. Harjunen.

Setzer, V.W & Carvalheiro, F. (1993). Algorithms and Their Analysis - A Pedagogical Introduction (in Portuguese). Caderno da Revista do Professor de Matematica (Vol. 4, N° 1, pp. 1-26). Brazilian Society of Mathematics.

Setzer, V.W. & Hirata, R., Jr. (1990, May/June). The Computer Day: a compact introduction to computers and computing (in Portuguese). Ciencia e Cultura, 42 (5/6) pp. 333-340. Brazilian Society for the Advancement of Science.

Spock, M. (1978). *Teaching as a Lively Art*. New York: Anthroposophic Press.

Stebbing, L. (1962). *Understanding your Child - a Word from Parent to Parent*. Sussex: New Knowledge Books.

Steiner, R. (1964). *Kingdom of Childhood*. Hudson: Anthroposophic Press.

Steiner, R. (1982). *Roots of Education*. Anthroposophic Press

Talbott, S. (1995). *The Future Does Not Compute - Transcending the Machines in Our Midst*. Sebastopol, CA: O'Reilly & Associates, Inc.

Thornburg, D. (1994, October). Keynote Address: ICUE Conference, Des Moines, Iowa.

Weinzenbaum, J. (1976) *Computer Power and Human Reason - from Judgment to Calculation*. S.Francisco: W.H.Freeman.

Winner, L. (1993, August 4). How technology reweaves the fabric of society. *Chronicle of Higher Education*, 36, B1.

Lowell Monke is assistant professor of education at Wittenberg University in Springfield, Ohio. Prior to that he taught young people with and about computers for 15 years, the last eight of which he spent teaching Advanced Computer Technology for Des Moines Public Schools. He is co-author of Breaking Down the Digital Walls: Learning to Teach in a Post-Modem World, which will be published by SUNY Press in January 2001.

Valdemar Setzer is a retired (but still active) full-professor of Computer Science at the University of São Paulo, Brazil. He is the author of scientific papers and books on compiler theory and construction, databases, software engineering, education in computing, as well as the book Computers in Education, Floris Books, Edinburgh 1989, with editions in Brazil, Germany and Finland. His web site, www.ime.usp.br/~vwsetzer contains numerous essays and papers on education and social and individual impacts of technology in English, Portuguese and Spanish. Besides being a public lecturer on these topics, he is a technical consultant for various firms in his fields of specialization.

This work was supported by the State of Sao Paulo Research Foundation grant 93/0603-1.

