MICROWARE

By Leonard F. Proctor

If you have suggestions or contributions that you would like to make to this column, please forward them to: Dr. L.F. (Len.) Proctor, Dept. of Educational Communications, College of Education, University of Saskatchewan, Saskatoon, Sask. S7N 0W0.

Microcomputer Acquisition Considerations

by M. Laucht

The acquisition and implementation of a microcomputer system involves many considerations beyond the purely technical details described in manufacturers' brochures and in the literature. This paper will outline some of these considerations.

User or Hobbyist?

Purchasers of microcomputer systems may be characterized as either "hobbyists" — those with a desire to own a microcomputer, or as "users" — those who feel that a microcomputer system may meet some specific needs. The distinciton between hobbyists and users is a necessary one since each has different expectations of, and requirements for, a microcomputer system.

A hobbyist is essentially free to choose, from among the many microcomputer, systems available, any system which strikes his fancy. The selection process need not be particularly rigourous. In fact, if a poor choice is made, the cost to the hobbyist is unlikely to exceed the actual purchase price of the microcomputer system.

A user must be more careful in his selection. The incidental costs of not meeting the need for which the system was purchased may greatly exceed the mere purchase price of the microcomputer system.

* An earlier version of this paper appeared in the **Manitoba Science Teacher Volume XXIII**, Number 3, Summer 1982 under the title of "Microcomputer Pitfalls".

This paper will dwell on considerations of particular importance to users, rather than hobbyists.

The microcomputerization process as a whole, irrespective of the particular application involved, consists of:

- setting goals and objections for the system
- determining various implementation options
- evaluating the implementation options and selecting the optimal one
- implementing the selected option.

1 The setting of goals and objectives should occur prior to any detailed examination of implementation options, and certainly must occur before any evaluation takes place. Otherwise not only is the system design and selection rather difficult but those expectations which are then left unspecified, may be unclear or unrealistic. Such unspecified expectations very often remain unmet, resulting in dissatisfaction and frustration.

2 Once the goals and objectives have been set it is possible to determine the various implementation options. Among those to be considered should be non-microcomputerized options, possibly for example, time-shared access to a larger computer. Certainly manual, non-computerized solutions, including relatively simple procedural changes should also be considered. And yet another is the "do nothing" option — that is, to continue with the present system.

3 When evaluating the implementation options the usual criteria i.e. costs, functionality, ease of implementation, ease of use, etc., are likely to be important. As well though, it is useful to avoid being dazzled by the new, emerging technologies and to bear in mind that what a particular system can potentially do, and what it actually does do, may be several years apart.

4 Of particular importance when dealing with computerized systems of all sorts is that people, procedures and materials are not the only components. Also present are the two components of all computerized systems: the hardware (equipment), and the software (computer programs or instructions). If either of these is inadequate, or unreliable, the system will not function properly. In education, the latter, that is the software has received most attention.

5 While the capabilities and features of microcomputers and related hardware have increased in recent years, and quite dramatically so, and while the microcomputer of today might be as powerful as the minicomputer or mainframe computer of yesterday, it is not as powerful as the minicomputer or mainframe computer of today. The same advances which have so improved the price/performance ratio of microcomputers have improved it for other sizes of computers as well. A microcomputer cannot be expected to do everything that a larger computer can do.

6 Any computer (micro, mini or mainframe) will have limitations. Therefore, it is important to match a computer's features and limitations with the requirements of a particular application. If this is done the limitations should not be much of a problem. If, on the other hand, a purchase is made too quickly, and with too little forethought, then the whole process can be a discouraging and frustrating expeience.

7 When evaluating a particular manufacturer and his line of microcomputers, it is important to consider issues relating to the long term use and development of the computerized application. As needs evolve, the microcomputer system should evolve as well. Such evolution includes the purchasing of additional hardware (more memory, auxilliary storage, peripherals etc.), obtaining more software or upgrading to a larger computer. It is desirable to do all of this with as little impact on the operation of the system and as little modification to hardware or software as possible.

8 Another consideration is hardware reliability. Reliability data for microcomputers is not always easy to obtain, but many very well be worth the extra effort required. Polling users of a given microcomputer system may provide valuable insight. As well, since no manufacturer's products are blessed with an immunity to failure, it is desirable to ensure that high quality repairs and parts are available locally, quickly and at reasonable cost. If not, then it becomes necessary either to wait until equipment is repaired, or to make appropriate backup arrangements. The need for a good repair facility should not be dismissed too lightly.

9 Also of concern is a microcomputer manufacturer's reputation and future prospects. The longer the company has been in operation and producing microcomputers, the better the reputation of the product, the greater the variety

Continued on page 21.

BOOKS

Personal Computers for Education
Alfred Bork
New York: Harper & Row, 1985.
Reviewed by Earl R. Misanchuk
University of Saskatchewan

Alfred Bork, a pioneer in the use of computers in education for nearly two decades, has written a refreshingly different book about computers in education. It is intended for a wide audience: teachers, university professors, educational planners and administrators, parents, and trainers. Bork's espoused purpose in writing the book was to provide a more balanced view of computers in education than available in other introductory books on the topic; I think he succeeds. In fact, I would put this book on the "must read" list for every educator who is involved in any way with computers in teaching, as well as for those educators who know nothing of computers yet, but feel they should. It should prove to be equally useful at all levels of education, from pre-school to university and continuing education.

Chapter 1 provides a good overview of the contents of the book, and suggests a reading sequence of only six of the 14 chapters for those who want to get the main points of the book without reading all 179 pages. Bork's writing, however, is so readable, so down-to-earth, so obviously written by an educator who has seriously considered the educational requirements of computers — rather than the gadgetry-oriented pipe dreams that frequently get published under the guise of introductions to computer literacy — that it is difficult **not** to read the whole book. To add further "spice", throughout the book are interspersed statements that those who are currently involved in promulgating computer literacy in the schools just might not like. A list of his comments critical of common practice would include, for example, that knowing how to write programs is not necessary to attain computer literacy; that high school curricula (and corresponding teacher education courses) which teach a variety of languages, instead of teaching good programming style with — at most — one language, are doing the wrong thing; and a strongly-worded admonition not to teach (or learn) BASIC, which he calls the "junk food of computer programming languages". He doesn't just offer them as opinions, either — he presents credible arguments to support them.

What distinguishes **Personal Computers for Education** from the horde of introductory computer literacy books for teachers is, in part, the arrangement of the chapters in the rest of the book, which is divided into three sections: **Modes of Computer Usage, Further Details About the Computer,** and **The Future.** The emphasis is on educational concepts, not on computers and computing concepts; in particular, the emphasis is on **learning** and computers. Reference to hardware does not come until Chapter 9, and even then, the presentation is conceptual, with minimal reference to bits, bytes, CPU's and all the other jargon that supposedly characterize the **computer literate.** Nowhere is there a chapter (or even a section of a chapter) entitled **The History of Computing!**

The first section consists of five chapters: **Learning to Program, Computer Literacy, Intellectual Tools, Computer-based Learning,** and **Management Systems.**

In Chapter 2, Bork presents a straightforward and concise description of the various elements that lead to good programming style; an explication of why one might want to learn how to program (while pointing out that the skill might not be necessary for the achievement of computer literacy); and a description of various approaches for learning a programming language, indicating that the most common method is not the most efficacious. A theme, picked up later in the book, emerges in this chapter: students can learn much from one another, and such peer learning should be encouraged in the learning of a computer language, since teamwork is common in the real-world application of the skills.

Chapter 3 deals with that elusive construct, computer literacy. After offering his own definition (which many will find less demanding than most other definitions, in terms of what one needs to know in order to be called computer literate), Bork lists (and justifies) the ingredients he considers necessary: social implications of the computer; common applications (word processing, spreadsheets, personal data systems, and other, area-dependent, applications); aspects of programming, particularly style; and a critical attitude toward computing. He quite deliberately omits hardware and extensive programming experience, and suggests that often, **no** exposure to programming is better than some, especially if that some involves BASIC. He concludes the chapter with a model curriculum for computer literacy spanning preschool to university.

Intellectual Tools, the fourth chapter, illustrates just how the computer can be applied as a tool in various subjects. Word processing, spreadsheets, paint programs, simulation languages, graphics, and electronic mail are illustrated in use. A concise, yet very understandable, introduction to Logo should give the novice reader sufficient understanding of that language to see how powerful concepts, such a top-down programming, procedures, and recursion, can be taught to even very young students. Bork points out Logo's warts, as well: it doesn't require the user to carefully define data types, and it emphasizes the logical side of programming at the expense of the data side. Despite the shortcomings, Bork bemoans the fact that few teachers ever get past the introductory, graphics-generating stage of learning Logo, as it is capable of performing quite well as a full-fledged language.

Chapter 5, **Computer-based Learning,** again reflects Bork's emphasis on learning, rather than teaching, and returns to the theme that students needn't always work as individuals: a great deal of very important learning takes place in the interchange among students when they work with a computer in a group. Bork describes and illustrates several examples of computer-based learning, including arithmetic drill and practice; sentence combining; some programs intended for young children; a physics program entitled **Batteries and Bulbs,** which he and his colleagues developed at the Educational Technology Center, University of California, Irvine; an electronics trouble-shooting program; "controllable words" programs; a logic course; a medical education course; and a spelling/reading program.

Testing, sequencing of materials, student databases, teacher aids (such as grade-books), and student aids form the content of the chapter entitled **Management Systems.** Another theme that is carried through the book — that multiple-choice questions, whether in instruction or in testing, are an obsolete technique that should never be used in computer-based learning — emerges here. Examples of the record-keeping systems used in a couple of computer-based learning systems are presented.

The second major section, **Further Details About the Computer,** contains the chapters **Advantages and Disadvantages of Computers in Education, Teacher Evaluation of Computer Materials, Hardware — The Raw Computer, Practical Issues of Computer Use in Education,** and **Developing Curriculum Materials for Computers** I and II (two chapters).

Chapter 7 begins with a Verse and somewhat simplistic overview of three different theories of learning, which could leave the non-professional reader believing that teachers and curriculum materials designers actually fall neatly into one of the three schools of thought, rather than being somewhat eclectic in this regard. The advantages of computers cited, however — learners enjoy computers; true individualization, interaction, and visualization are possible; and learning occurs more rapidly with computers than with conventional instruction — as well as the disadvantages identified — there is very little good educational material available; few teachers have adequate knowledge of how to use computers effectively; the potential for problems vis-a-vis unauthorized access to student databases; and potential educational disparities based on affluence and accessibility to computers — are well covered.

Teacher Evaluation of Computer Materials is a chapter that should be read by anyone presuming to select educational computer programs. Bork lists a number of features of programs that should be evaluated critically (content, environment, approach, interaction, user-friendly interface, individualization, graphics, screen design, and sound) and gives another list of features that should count against any materials being evaluated (books via computer, dense screens, and multiple choice interaction). This chapter should be successful in its espoused purpose of "helping teacher **(sic)**, administrators, parents, and others to judge what computer-based learning material is satisfactory and what is not (p. 94)".

Chapter 9, **Hardware — The Raw Computer,** is in some senses the weakest in the book. While accepting that Bork's intent was not to provide great detail about hardware, I feel he gives some very important considerations short shrift. His message appears to be "buy for the future" when acquiring hardware; he claims that there is so little good and useful software currently available that one should not be impressed by salespersons' arguments that much educational software exists for certain machines. Instead, says Bork, one should buy "state of the art" machines, which he equates with 16-bit computers (despite the fact that the book carries a 1985 copyright date, there is no acknowledgement of the existence of 32-bit machines). He seems to assume that quality educational software will necessarily follow for those machines. Furthermore, he makes a considerable error of omission in not specifically pointing out that some of the wonderful programs he has described earlier in the book do not now, and probably never will, run on the computers he seems to recommend due to inter-machine incompatibility and lack of standardization, a topic he ducks completely. His comments on what to look for and think about when purchasing machines are too general to be helpful.

Practical Issues of Computer Use in Education deals with the various possible school and university environments for computers, and reminds the reader that a great deal more planning must go into the acquisition of computers than a decision to "get a couple and see what happens". Physical location and maintenance are just two of the concerns that must be addressed.

The two chapters entitled **Developing Curriculum Material for Computers** begin with a recounting of the process of producing a textbook; this is followed by an explication of the steps leading to computer-based curriculum material (pedagogical design, graphic design, implementation, evaluation, and improvement). The point becomes obvious: It is essentially silly to believe that a teacher, working evenings and Saturdays, can acquire the necessary skills and execute all phases of preparing computer-based learning materials, and come up with a good product. Authoring languages or systems, Bork contends, are no help, because they unnecessarily limit an author's capabilities; they lead to the development of poor materials. Even given sufficient time, for an individual to perform all phases of the process requires such a rare combination of skills that to recommend this model of development is unrealistic, suggests Bork. A detailed example of a small segment of a pedagogical design (the analog of a storyboard) illustrates fully what computer-based learning can be. For me, this was the most exciting part of the book; at the same time it was sobering, because it illustrates so profoundly the amount of creativity and work that must go into just a minute's worth of student learning time. Sections on screen design, coding, and review, evaluation and revision (the later with explicit criteria for all phases of the process — a valuable checklist!) illustrate just how specialized the roles of development team members have become.

Particularly of interest to me (reading the book in February) was his description of where most of the pedagogical design is actually done by the Educational Technology Center group: in the mountains of Southern California, at pool-side, or in a Jacuzzi!

The preparation of good computer-based learning materials continues to be an expensive process. Bork gives an example in which two hours' worth of program time costs $25,000 to develop fully. It seems clear that educational technology, in the face of the computer age, must give up one of its long-cherished philosophical cornerstones, local production. "Quality costs money" seems to be the message.

He rounds out the chapters on developing materials with eight points so germane that I repeat them here verbatim:

(1) Educators must insist on high quality material; it should be highly interactive, not booklike or lecturelike. Such simple-minded tactics as multiple choice should be avoided.

(2) Educators must be critical of existing material — most of what is available is junk.

(3) The test of any production system is in the quality of the product. Thus abstract arguments about production systems are not critical. Rather, in each case the material produced should be examined.

(4) Computers will become more valuable in learning as educators move from strictly behavioral viewpoints to viewpoints that take into account cognitive psychology.

(5) Group effort is needed to produce effective curriculum material. Most educators seem to agree with this, but many are still implicitly using tactics that derive from older nongroup strategies.

(6) Authoring languages and systems should not be used in the production of computer-based learning material. In most cases they lead to uninteresting or trivial material. However, authoring tools can be useful in the coding process.

(7) Far more experimentation in the development of computer-based learning material is needed. The best curriculum production systems can undoubtedly be improved.

(8) Educators' strategies need to be future-oriented, particularly toward the largescale production that will be necessary in the next few years.

The final and shortest section, **The Future,** contains chapters on **Future Computer Equipment,** and **Computers and the Future of Education.**

Chapter 13 is another weak one, and may readily be skipped. On the one hand, Bork becomes a little "pie-in-the-sky-ish"; on the other hand, he reports on "new" and "future" developments in chip technology and software that are already out of date, due to the time lag inherent in the publication of a book. Indeed, a few of his predictions, like his obviously biased view of the popularity of Pascal, have turned out to be, arguably, wrong.

The final chapter, however, should (must?) be read by all. He begins by making a bold prediction: computers will become the *dominant* (italics mine) delivery system in education at all levels. Citing the current problems education faces — decreasing favor and credibility among taxpayers, coupled with severly reduced budgets — he gives examples of how private enterprise is already moving in to fill the gap. It is not a pretty picture that he paints, but anyone at all concerned with education should take a long and hard look at it. Of course, Bork does offer up a hope of salvation: Not surprisingly, it is the increased use of computers to teach. He paints a compelling scenario in which computers do the content delivery, while teachers act as the ultimate source of help, and as socializers. I know, we've heard that song before — during the '60's, when the newly-discovered systems approach was to have married with educational media (especially television) to save education. It didn't work. But times are different now; and so are the tools. Maybe, just maybe, the availability of the power of the computer, coupled with societal discontent with the current state of education, will bring about the much-needed reform this time.

MICROWARE

Continued from page 18.

of accessories and software available, and the more fiscally stable (or growing) the company is, the more likely it is to remain in business. Or is it?

10 A user should, as much as possible, avoid becoming stuck with a technical orphan, that is a microcomputer whose manufacturer has gone out of business or has stopped supporting it. This means, that parts, repairs, accessories and software are likely to become increasingly difficult or expensive to obtain.

11 Software is as important as hardware. No matter how good, reliable or cheap a particular hardware configuration is, it is of little use if the required software, is not available or cannot be written to the functional, time and fiscal requirements imposed by the given application.

12 With every decrease in price for computing equipment, a new community of users and programmers is created. Each such community follows the same pattern. Software is initially unstructured, ad-hoc, unreliable, difficult to debug and difficult to maintain. As computer use by this community matures, the software development process also matures, resulting in more disciplined, more structured, more modular software which is easier to debug, to use and to maintain.

The design and development of such software is a complicated process involving the resolution of contradictory requirements by means of compromises and tradeoffs. Thus while programming was once an art, as practiced by professionals today, it is a discipline. If at all possible a user should avoid attempting to do this himself without adequate training.

Most people would not consider building their own hardware, yet many of these same people will seriously consider designing, creating, debugging, documenting and maintaining their own software. Great care should be exercised before taking this particular course of action.

13 When purchasing software there are essentially two options: custom-written software or ready-made, off-the-shelf software.

With custom-written software it is important to evaluate the software suppliers as carefully as the hardware suppliers. Programming a system which is to run correctly, effecitvely and reliably and which is to be maintainable (easily modifiable when changes are required), is a discipline requiring a methodical approach including analysis, development, debugging, testing and documentation. It is wise to ensure that the source of custom written software has the requisite training and experience.

It is especially important to beware of the technical hotshot. This is a programmer who can compress program instructions into the smallest possible amount of memory, who can write programs which execute exceptionally quickly or who talks incessantly about obscure technical details. Such experts are needed in specialized circumstances but not for most applications. Far more important than a program which is faster or more compact is a program which does what is required in a reasonably simple and straight forward, user friendly, efficient, reliable and maintainable a fashion as possible.

While having software custom written should provide a user with exactly what he specifies, he will also be getting software which is, by definition, a technical orphan. He faces essentially the same risks as for hardware technical orphans, especially potential problems in updating the software and in finding latent bugs.

In buying ready-made, off-the-shelf software, a user stands a better chance of getting reasonably priced, working software. It is also possible that upgrades and enhacements will be available. If a software supplier is selected carefully, buying software ready-made can significantly reduce risk and cost.

The price the user pays for this reduced risk is that he has to take what is available. While some customizing of off-the-shelf software is sometimes possible, there is never quite the freedom to specify the application as when the software is custom written.

No matter how the software is obtained it is important that it is of high quality, that it is easy to use, correct, reliable, and of course compatible with the hardware to be used.

A major concern about the quality of software is correctness. For an educational environment the correctness of software is no less critical than for other application areas. Because complete testing of programs is frequently difficult and often impossible, great care should be taken in selecting a source for software.

It is further necessary to ensure that the techniques used both for CAL programs and for other computer/student interactions are pedagogically sound.

Future evolution of the system should be kept in mind but one shouldn't get carried away and constantly change the functional requirements in an attempt to add every bell and whistle as one thinks of it. This will unduly delay the system's completion.

14 Beware of bargains. Good software is expensive and very time consuming to produce. Inappropriate or poor quality software is too expensive no matter how cheap it appears.

15 If compatible software and hardware are available from a variety of suppliers, so much the better. A great deal of "plug compatible equipment" (equipment which may be easily plugged into a system instead of the equipment originally intended for that system) and many software packages are now available from independent suppliers for many different microcomputers. As long as one is careful to ensure that what is bought is in fact compatible, such diversity of suppliers opens up many options, often at reduced cost.

One should be especially careful when purchasing a microcomputer simply because some other, preferred approach for the application is too expensive. The initial purchase price of the microcomputer is only one of the potential costs. Operating costs, software development, debugging and maintenance costs, hardware repair costs, hardware and software upgrade costs, and the cost of delays inappropriate system components etc. are much more difficult to calculate and are likely to be much more significant in the long run.

Microcomputers have great potential for education but as with every tool, one must select and use them carefully and appropriately. If one considers very seriously the potential pitfalls of (micro) computerizing an application as well as the potential benefits, as listed above then chances for a successful implementation will be greatly improved.