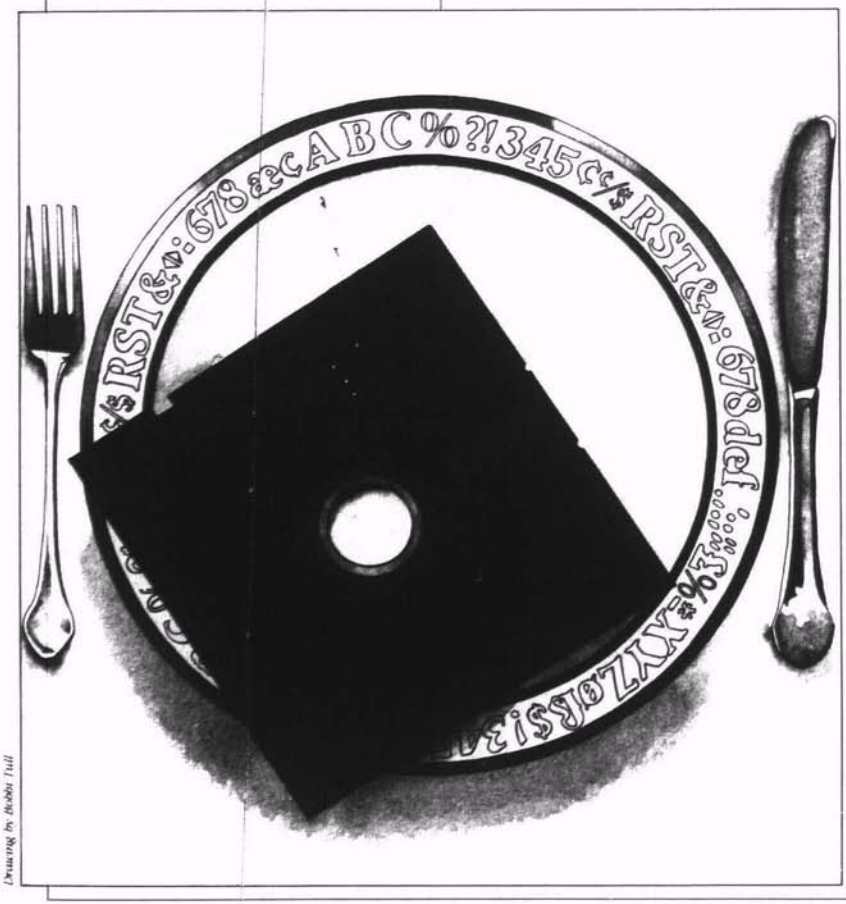
Developing More Powerful Educational Software

Schools can use computers more effectively by assigning them a primary role in delivery of instruction and by applying sound educational theory in software design.

JOHN HENRY MARTIN



Drawing by Bobbi Tull

Schools as social institutions have an organic-like capacity to ingest and convert technology to institutional purposes, often with severe consequences to the media's effectiveness with students. Since 1900 schools have absorbed typewriters, motion pictures, radio, television, language laboratories, teaching machines, and a lesser group of visual aids such as overhead projectors, tape recorders, and filmstrips. In the last five years the computer as an instructional device has been added to that long list, and signs are growing that it may suffer the same fate.

We need to analyze the reasons for this seeming imperviousness and determine how schools can take full advantage of the extraordinary power of this new technology, which in many ways incorporates all the others.

It is important to recognize that the potential value of a new technology is no guarantee that it will be realized. Thomas Edison wrote in 1897 that his motion picture machine would eliminate the need for books. In the 1920s David Sarnoff thought radio would bring about radical changes in schooling. The Ford Foundation spent millions promoting educational televi-

sion. Paradoxically, while motion pictures, radio, and television have become giant industries with massive effects on our culture, their contribution to teaching and learning has been minimal.

Reasons for Resistance

The minor impediments to effective use of technology in schools are easy to isolate: the high cost of equipment, the need for training, the lack of fit between new and existing materials, and burdensome logistical problems, including storage, availability, and safety. The strongest institutional factor, however, is the almost universal assumption that technology is *only a supplement* to the teacher and the text. The concept that films, radio, television, or computers might serve as a *primary delivery agent* for instruction seems rarely to have survived beyond the inflated rhetoric of a technology's earliest days.

In keeping with this conception of technology we have catalogs of available films and videocassettes carefully listing their linkage to parts of the existing curriculum in a wide variety of subjects. These topical segments are typically confined to a 40-minute or less exercise. Usage is intermittent, idiosyncratic to the teacher, and locked into the purposes and content of existing courses of study. In other words, the manner in which the technology has been used by the schools has *predestined it to perform only a minor and subordinated role*.

These limiting factors, inherent to the nature of schools, have been augmented by systems of belief derived from the perceived nature of computers. When designing software for business use, the common practice has been to analyze precomputer procedures, convert the analysis into a flow chart design, enhance the original procedures by virtue of the speed and data handling capacity of the computer, and then program that design into the computer.

Systems engineers turned from this business model to the world of education naively expecting to find the relatively well-defined practices used in business paralleled by equally specific procedures in the educational world. Unfortunately, the aspect of instruction most easily adapted to computer delivery was the drill-and-practice workbook. The few programmers who turned to learning theory found little agreement and much fragmentation. Only Skinner seemed to have an organized, readily adaptable schema. The results of this approach were indistinguishable from drill-and-practice exercises in print, although their poor quality was partially camouflaged by the dynamic appeal of the computer to students.

Another factor contributing to unimaginative educational software has been the inherent power of the computer to perform functions that appeal to the institutional needs of schools for recording test results and tracking student progress. These necessary bookkeeping functions of the school are part of a well-designed structure of learning, but they constitute the frame, not the picture.

To make matters worse, the extraordinary rise and fall of computer games induced designers of educational software to add a layer of game-like characteristics to many of their drills. Clearly, the natural appeal of games should not be ignored; integrating game theory with the content and skills to be taught has a synergistic effect. Chance and risk, along with graphic evidence of growing skill perceived by the participant, are strong reinforcers. Challenge and humor can be effective lubricants to learning. Nevertheless, covering the stale bread of dull materials with a confection of gaming has not made an educational cake.

Another consequence of the hardware-dominated view of educational software has been the more than 20-year effort to produce course-authoring programs. The hope was to give "authors" all the computer routines, visuals, screen formatting, and student management protocols needed to construct a computer learning program in any subject. From the first crude efforts pioneered by IBM in 1964 to the increasingly more subtle and dexterous systems now available, the results have been mixed. There has been a steady growth in the sophistication of the routines available to users. However, the approach has restricted the creativity of designers by forcing them to conform to the programs' built-in psychological concepts, which with rare exceptions remain unnecessarily primitive.

Sadly, course authoring programs have encouraged the shallow notion that every user of such a system could be an educational software developer. In some respects promulgation of such programs has been like the medieval search for the alchemist's stone: focusing our attention on the wrong chemistry of learning. Rather than limiting the design of software in this way, we should have been exploring the implications of more ambitious conceptions of the learning process.

Toward Theory-Compatible Software

There is too little space here for a full exposition of how each learning theory fragment we have can be engineered into software designs. In the interest of clarity, however, and because the need is urgent, I will describe one such design resulting from synthesis of an observation by Piaget with a separate thesis advanced by Bruner.

Piaget wrote of children's efforts to impose their pre-existing "cognitive structures" on their environment. If they succeed, he said, they will revise those structures. Bruner sharpened our awareness of how learning could be facilitated when, 25 years ago, he recommended that curriculum designers bring out the essential intellectual structure of the subject to be studied.

Taken separately or together, these generalizations are not readily transmuted into designs for the improvement of learning. Bruner's intellectual designs can be too deceptively simple, as well as too deeply embedded, for all but the best of scholars to discern. Piaget's "cognitive structures" are not visible; they can only be inferred from careful observation of repeated patterns of behavior.

With patience and insight, however, these patterns can be detected—and

applied in the design of educational software. For example, kindergarten and nursery school age children will repeatedly pile similar items into a teetering column, which eventually collapses. Eventually the child will accept a horizontal row as a satisfying result, constructing teaspoon paths across the kitchen and plastic roadways around the classroom.This propensity to "line things up" seems to be an outward manifestation of an inner "cognitive structure."

Bruner's thesis required that for our purpose (helping children learn to write and read) we needed to look for the intellectual design fundamental to that task. Specifically, English words are written in an alphabetical code that is read from left to right, with each letter or letter combination representing the exact sequence of sounds made in speaking those words. This lining up of letter symbols for the sounds of speech can be linked to children's tendency to line up physical objects. What was required of the computer exercises was to help children make closure: to "see" (frequently at the level of cognition but often at a subliminal level of awareness) a match between their disposition to line up teaspoons and the lining up of letters in a word.

The IBM computer-centered "Writing to Read" system embodies this design. We show a drawing of a cat on the computer screen, use digitized speech to ask the child to say "cat," and then bring the word *cat* to the screen. Next we remove the drawing and put the word back on center screen. The letter *c* is moved slowly to the upper edge of the screen, then letter *a* is moved to the upper left corner of the screen, and the letter *t* to the lower edge, so that the three letters are in alphabetical order. The computer asks the child to say "cat" again and, as though responding to the child's command, the letter-sound symbols march in appropriate order, one by one, to *line-up* as the word *cat*.

I have provided this example to illustrate the architecture of one educational design resulting from the engineering of learning theory into educational software. This particular architectural design is generic to learning and can be used again and again in constructing educational software. For example, the human heart can be shown in graphic simplification as a four-chambered pump with entry and exit tubes in appropriate valves. The flow of blood into one auricle, squeezed into the ventricle, then into the lungs for oxygenation, returning to the second auricle, and eventually exiting as fresh arterial blood is once again a sequencing: a "lining up" cognitive structure. Such a graphic, accompanied by a soundtrack recorded from a stethoscope synchronized with the moving chart of an electrocardiogram, should unquestionably foster learning about cardiac functions. The same architectural design can produce superior computer-based programs to train automobile mechanics to understand the function and repair of carburetors.

This software design seeks to match the patterns of thinking already present to simulate the patterns of structure and movement inherent in the object of study. We need to recognize that this conscious effort of the learner will be augmented by "complex processes which take place below the conscious level of the mind-processes which permeate the way we think but of which we are unaware" (Hofstadter 1980, p. 569).

Creative software must also accommodate different types of learners. With some children we can see the eureka phenomenon: the flash of insight that reveals sudden understanding. With others comprehension seems to result from slower incremental growth: a process whose elemental parts learners cannot fully articulate, but whose functioning they can demonstrate. For them learning seems to occur through slower subliminal understanding. We are tongue-tied, for instance, trying to describe how we tie a bow-knot on a shoe, yet our fingers move with the unconscious dexterity of complete mastery.

The Threat and the Promise

The evidence is growing that schools are becoming increasingly disenchanted with the sad condition of present software (Dronka 1985, Tucker 1985). Teachers aware of the rise and fall of educational fads from modern math and science, to the open classroom, to teaching machines, language laboratories, and television have learned a healthy skepticism that is beginning to surface toward the computer as an instructional tool.

At the same time, new developments in technology are making the promise of computer-based education even more exciting. Expert systems will soon expand the power of computers exponentially even as scientists continue to home in on the intricacies of the human brain. These forces are likely to converge and culminate in an artificial intelligence that will bring an extraordinary increase in the power of educational technology. And as industrial trainers rapidly adopt the computer-controlled videodisc, schools will be challenged to make imaginative use of this highly flexible learning technology.

While these new imperatives may override many of the obstacles and errors I have described, history tells us that the growth in capabilities of the technology will not by itself assure a corresponding increase in the sophistication of our crucially important professional knowledge. If schools are not to become poverty-bound nonparticipants in another sweeping change, we need to press on toward higher quality software that creatively and richly applies the fragments of learning theory that we already possess.□

References

Dronka, Pamela P. "Computer Integration into Instruction Is Stuck . . ." *ASCD Update*, Summer 1985, pp. 1, 6–7.

Hofstadter, Douglas R. *Godel, Escher, Bach: An Eternal Golden Braid*. New York: Random House, Vintage Books, 1980.

Tucker, Mark S. "Computers in the Schools: What Revolution?" *Journal of Communications* 35 (1985):12–23.

John Henry Martin is president, JHM Corp., Carlisle Castle 207, 1850 Palm City Rd., Stuart, FL 33494.