SOME NEW (AND OLD) DIRECTIONS
FOR COMPUTER COURSEWARE

J.D. Fletcher

CSE Report No. 234

1984

CENTER FOR THE STUDY OF EVALUATION

Graduate School of Education

University of California, Los Angeles

The project presented or reported herein was supported pursuant to a grant from the National Institute of Education, Department of Education. However, the opinions expressed herein do not necessarily reflect the position or policy of the National Institute of Education and no official endorsement by the National Institute of Education should be inferred.

TABLE OF CONTENTS

In 1960 T.F. Gilbert wrote:

> If you don't have a gadget called a "teaching machine", don't get one. Don't buy one; don't borrow one; don't steal one. If you have such a gadget, get rid of it. Don't give it away, for someone else might use it...This is the most practical rule, based on empirical facts from considerable observation. <u>If you begin with a device of any kind, you will try to develop the teaching program to fit that device</u> (p. 478.)

Gilbert's point of view is one with which many of us will sympathize. Educators who have mastered their craft through considerable investment of time and energy in learning how to use the traditional technologies of text, lectures, blackboards, and real-equipment laboratories have every right to be suspicious of new technology that threatens to revolutionize the hard-won techniques now at hand. Even programmers, initiates into the priesthood of computer technology, are occasionally elevated by computers to levels of frustration in which they are willing--and eager--to destroy thousands of dollars worth of equipment with their bare hands. Moreover, Gilbert is undoubtedly correct when he suggests that we may develop teaching programs to fit the technology at hand. Of course we will, and to varying degrees we always have. To suggest that we should not pursue new technologies for this reason may not be so correct.

As Marshall McLuhan (1967) pointed out, every technology, to some extent, carries its own message. To ignore this message is to neglect the strengths of the technology. The technologies now becoming available will not only provide powerful new instructional tactics for presenting context, they will also make some content accessible that heretofore could not be taught in any practical setting. In the development of computer courseware it is possible to discern entirely new "functionalities" in instruction. As is true of most technological efforts, we have begun by trying to enhance the capability of our existing practice. We may end with new capabilities that change the nature of what we do in ways that are completely unanticipated. This could be the essence of the new computer revolution in schools. It is not just that we will have computers everywhere or that we will enhance our capabilities to instruct. We may also change our ideas about what instruction is. Not only will we get better at doing what we do now, but in a fundamental sense we may change what it is we want to do.

NEW DIRECTIONS

It may be well to begin with a fable. This fable will already be familiar to some readers. Nevertheless, it seems sufficiently relevant to bear repeating. As the story goes, there once was a government "blue-ribbon" commission of instructional experts assembled to specify the ultimate in instructional technology. After several several days of meetings--suitably fueled by long lunches and accommodated by comfortable lodging--the experts came up with the following specifications for the new technology:

1. There should be no exotic power requirements. The technology should use ordinary household current, or be battery powered, solar powered, or require no power at all to operate.

2. It should be light and easily portable. One person should be able to transport it, and at best it would be carried in one hand.

3. There should be no complicated installation or environmental requirements. It should be easy to set up and use, it should operate in moderately extended temperature ranges, and it should be, as the military says, "ruggedized."

4. It should provide random access to a large amount of material.

5. It should be capable of displaying graphics, photographics, drawings, color, and high quality, easily read text.

6. It should be inexpensive, costing less than $50 a copy.

The commission report was received with great relief for, as the perspicacious reader may realize, no research and development money was required to develop the technology. In fact, the technology already existed and had been in place for over five hundred years. The appropriate technology was, of course, a book.

This is a fable for all of us in the business of applying new technology to instruction. We must come up with solutions that promise real innovations; in the case of instructional technology, they must be better than books. At the same time, some of our prototypes will be, like the horseless carriage, less efficient than what they are intended to replace.

Books are important because, among other things, they are able to capture instructional content and make it inexpensively available to an unlimited audience. As Bunderson (1981) pointed out, computer technology is important because, among other things, it makes both the content and the interactions of great instruction inexpensively available to an unlimited audience. This promise has yet to be realized, but it seems almost inevitable. What we need to do is sift through all the prototypal development and find therein those embryonic techniques that promise to be better than books. It turns out that these techniques are neither easy to find nor trivial to develop. I will briefly examine them in the three areas of drill and practice, tutorial dialogue, and simulation.

DRILL AND PRACTICE

"Drill and practice" is doubtless one of the more regrettable terms in instruction, evoking images of the classroom as a sweat shop and attracting the ire of those who want to use computers to create a rich and friendly learning environment for intellectual exploration and discovery in the classroom. Certainly it is now fashionable to deprecate drill and practice as a computer instruction technique, and it has been so for the last five years. Papert (1980) cites drill and practice as an example of the QWERTY phenomenon. It turns out that because the mechanical keyboards of earlier times were unable to keep up with skilled typists--the keys would jam and otherwise misbehave if they were operated too quickly--typewriter keyboards were originally designed to slow down the key presses of skilled typists. The result was the QWERTY keyboard, named after the topmost row of letters. This keyboard is with us today despite our having removed all the mechanical obstacles to fast operation that resulted in the QWERTY design in the first place.

Papert's argument is that early applications of computers to instruction necessarily followed drill and practice formats partly because that is what classroom teachers would accept and partly because the computer technology of earlier times could support nothing else. This point of view is not entirely accurate, as can be seen in the design of curricula for the IBM 1500 System in the mid-1960's. The Stanford beginning reading program is a case in point. This curriculum, which was designed roughly in the period 1964-1966 and is described more fully by Fletcher (1979), encouraged children to build matrices using words and spelling patterns, to read and to be read stories (with illustrations), to record and play back messages, and to experiment with linguistic forms and variations.

Teacher acceptance was an issue somewhat separate from the content and approach of the curriculum--using computers to teach at all and taking away from classroom time to do it were the central concerns of the teachers. Nonetheless it is notable that when the Stanford group moved to a less expensive machine configuration for presenting beginning reading instruction, the curriculum became more drill and practice in nature.

In any event, it seems past time to make a few arguments in favor of drill and practice. Is drill and practice an example of Papert's QWERTY phenomenon? The answer seems to be "no", partly because it works--drill and practice is still one of the most successful techniques we have in computer instruction--and partly because there is so much yet to be tried and developed in the drill and practice mode. Even if we assume drill and practice is limited to presentation of discrete items such as math facts or vocabulary items, there are at least three directions for curriculum development in drill and practice. These have to do with performance goals, optimal item selection, and optimal activity selection.

PERFORMANCE GOALS

We may best begin with trajectory theory. Basically this is a way of accounting for the progress, or trajectory, of individual students through a curriculum as a function of the amount of time they spend working in the curriculum. Figure 1 shows, perhaps more clearly, what trajectory theory is getting at. For individual students A, B, and C we try to predict and prescribe their grade placement on standardized paper and pencil tests based on the amount of time they spend on the computer curriculum. The interesting thing about trajectory theory is not just that it works, but that it has worked amazingly well in practice. In two published studies using trajectory theory (Suppes, Fletcher, & Zanotti, 1975 and 1976) the standard error of estimated grade placement was in the range .04 - .06 of a grade placement. In other words, the estimates were off by less than a tenth of a grade placement for 90% of the cases. Again, these estimates were based solely on the amount of time the student spent on the computer and were independent of what was being done in the classroom. If we want to predict and control progress toward measured goals of achievement, trajectory theory may be one of the best techniques we have. It is worth emphasizing that although trajectory theory was developed for drill and practice, it may be applied to any form of instruction where we have closely watched and accurate measures of time on task, as we have in computer instruction.

There are still many questions to be answered about trajectory theory. Can it be applied to all subject matter? Can it be applied to methods of instruction other than drill and practice? Are there

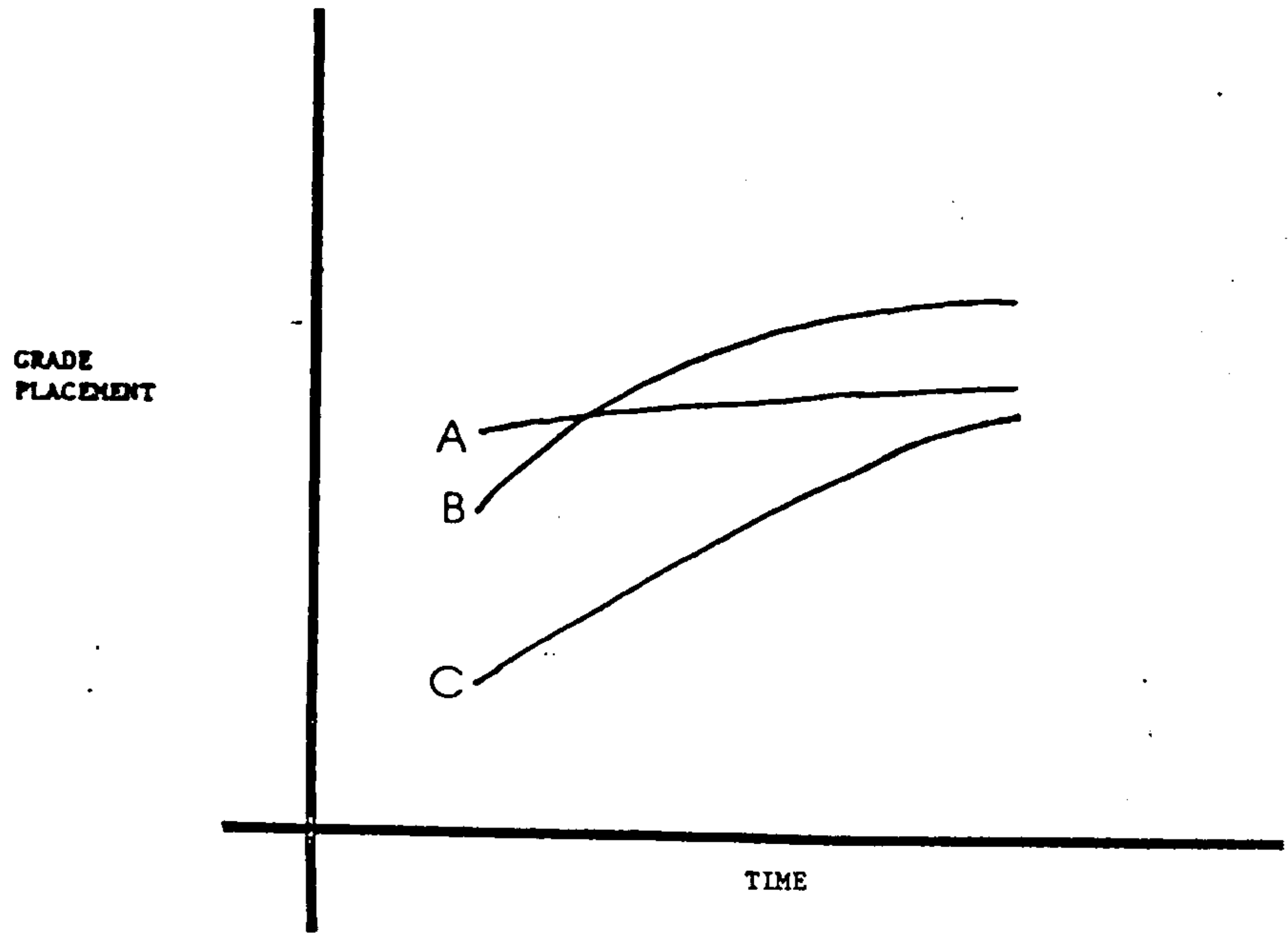

Figure 1. "Trajectories" of students through a curriculum.

significant and important benefits to be gained from using classroom observations of time on task as well as computer time to predict and control progress? The list of questions could be continued. Trajectory theory is not a particularly new technique for computer curriculum, but it remains promising and worthy of further development.

OPTIMAL ITEM SELECTION

Basically, an instructionally optimal solution is one that attempts to maximize some outcome, such as scores on an achievement test, subject to some constraints, such as total time on task, session length, and student ability. Optimal solutions are brought to use by control theory which in turn comes from operations research. It is a well known and noted fact that operations researchers tend to attack problems by removing from them everything difficult to quantify or measure and building an imposing mathematical structure on what remains. In the current instances, the imposing mathematical structures remain, but some portion of what is difficult to quantify or measure can be supplied by mathematical models of learning and memory. The wherewithal for applying both these models and control theory to instruction in real time is provided by computers in the context of computer instruction.

The problem of optimal item selection in instruction was stated with mathematical clarity and rigor by Suppes (1964), but can be stated fairly simply in words: given a large number of items to be taught and a fixed time in which to teach them, what subset of items should be presented to an individual student at a given time in order to maximize his or her eventual achievement? The answer can be supplied by the above-mentioned quantitative models of learning and memory. Figure 2 presents a probability state-transition matrix of an appropriate sort based on General Forgetting Theory (Rumelhart, 1967; Paulson, 1973). This matrix shows what can happen to an item when it is presented to a student. As can be seen from the figure, the model of learning postulated is very simple. If an item is in the learned state, it stays there. If it is in the short-term state, it will either advance to the learned state or stay where it is. If it is in the unlearned state, it will advance to the short-term state or the learned state or remain unlearned. General Forgetting Theory is actually a little more sophisticated than is described here in that it accounts for probabilities of correct responding separate from the learning status of items and, notably, it postulates what happens to the learning status of an item when it is <u>not</u> presented. An optimal strategy for item selection based on General Forgetting Theory is, like all models of this sort, fairly simple in its view of human learning but fairly complex to implement. It could not be implemented by a book.

Studies by Lorton (1973) for teaching spelling and by Laubsch (1970) for teaching foreign language vocabulary have shown approaches of this sort to be effective. They may even be dramatically effective, far more so than any other method for teaching large numbers of relatively independent items to students, but little work has been done in them since the mid-1970's. It seems to be a thread of research we have let slip through the cracks. There seems to be no

LEARNING STATE AT TIME T + 1

LEARNING STATE AT TIME T		LEARNED	SHORT-TERM	UNLEARNED
	LEARNED	1	0	0
	SHORT-TERM	c	1-c	0
	UNLEARNED	a	b	1-a-b

Figure 2. Probabilities of an item state transition when it is presented at time T.

real reason to drop it from our list of new directions for computer curriculum. Its promise for exceedingly efficient instruction remains.

OPTIMAL ACTIVITY SELECTION

A few words may also be in order for optimal selection of activity. This problem most clearly emerges in the context of "strands" approaches to curriculum development. The strands approach, which was first described by Suppes (1967), calls for the apportioning of a computer curriculum into various content areas, or strands. For instance, a curriculum in reading comprehension might be divided up into vocabulary, literal comprehension, and interpretive comprehension strands. The problem, then, for a computer curriculum designer is to decide how much time students should spend in each strand or, to state it a little more completely, how to control student progress in each strand so that each student's achievement over all strands is maximized at the end of some specified period of time. If progress in each strand is independent of progress in each of the others and if each of the strands contributes equally to the measure of achievement, then the solution is simple: we just pick the strand in which learning rate is greatest and allocate all the student's time to it. If, however, the situation resembled our reading comprehension example in which progress in one strand is interrelated with progress in the others, the situation is more complex. In reading, after all, a student with a poor vocabulary will not progress very far in literal or interpretive comprehension, yet the achievement measure of success for the curriculum will presumably be more concerned with comprehension than with vocabulary growth. Some sort of optimal mix of vocabulary development and work in comprehension will have to be devised for the student.

An appropriate optimal strategy (based on the Pontryagin maximum principle of control theory) for adjusting progress in interrelated strands was devised by Chant and Atkinson (1973) for the case of two strands. This strategy determines how much time a student should spend each day in each strand, depending on the student's learning rate in each strand and on how much he or she has already progressed in the strand. Extension of the strategy to curriculum environments with three or more strands was left by Chant and Atkinson as an exercise for the reader, but was described by the authors as being "straightforward". It very probably is, but it has not been done, or at least it has not been published. Moreover, there have been no applications of this strategy to determine in practice how much it really buys in terms of student achievement relative to other approaches. In other words, here is another promising direction which we have just begun to explore. It cannot be implemented in a book, and more needs to be done.

Most experimental psychologists reading the above discussion of drill and practice will find it difficult to suppress dark, uncomplimentary mutterings about "1960's psychology". There are cycles in research, as in most things. In this dimension, we seem to oscillate between attacking small, tightly constrained, and fairly uninteresting problems over which we exercise a great deal of control, and attacking very large, sloppy, and interesting problems over which we can exert very little control. As may be evident from the above discussion and from reviews by Atkinson and Paulson (1972) and Fletcher (1975), drill and practice emphasizes the former. Nonetheless, it should also be evident that drill and practice is not just a matter of throwing items at students who are treated in some assembly line fashion. There are deep, educationally significant, and scientifically credible issues yet to be settled concerning drill and practice. Finally, it should be evident that despite the early strong results we have had from drill and practice, much more could be done to fully realize the promise of this approach.

As far as the oscillation between tightly controlled, less interesting problems and poorly controlled but much more interesting problems is concerned, it appears that current research in psychology, applied psychology, and instruction emphasizes the latter. This trend is especially apparent in current attempts to build tutorial dialogue systems. Nowhere is the attempt to automate single tutor/single student dialogue more evident. This is the line of development to turn to next.

TUTORIAL DIALOGUES

Before diving into the area of tutorial dialogues, a few comments on the automation of programmed textbooks may be in order. Most commentators on tutorial dialogue approaches include in this category the intrinsic programming techniques of Crowder (1959) that appear so frequently in commercially available computer instruction materials. Basically this approach uses the computer as a very large and sometimes very intricately programmed textbook. This is an approach that could be pursued in a book, although the book might have to be carried around in a wheelbarrow. Nonetheless, this approach appears to concern application of book and text technology rather than computer technology to instruction. It remains one of the most common, easily produced, and frequently implemented approaches, and it is best supported by authoring languages for computer instruction. The development of authoring languages such as PILOT, TUTOR, WISE, PLANIT, etc., all seem to have intrinsic programming in mind since this is the approach most easily taken when one uses these languages.

We tend not to publish our unsuccessful attempts at computer instruction, among other things, but there seems to be an underground consensus among those in the business that these intrinsic programming aproaches do not work very well. What appear to be intuitively obvious and correct procedures for assessing student knowledge, deciding when to branch, and providing remedial and accelerated material turn out to be relatively ineffectual in the light of student performance data. The determined reader is welcome to peruse Fletcher and Beard (1973) as an example of unpublished--and unsuccessful--work of this sort. In any case, this section does not concern the automation of programmed textbooks.

This section is concerned with the development of intelligent instructional systems as a new direction for computer instruction. This approach is a direct attempt to imbue computers with the qualities of expert human tutors. This line of development grew out of early concern with just how long it took, and how expensive it was, to generate items for computer presentation. Early estimates of the amount of time required to produce one hour of computer instruction ranged from 77 to 714 hours on PLATO, 200-400 hours on TICCIT, and around 475 hours for the IBM 1500 Instructional System (Orlansky & String, 1979). One solution to this problem was sought by those who noticed that the process of preparing items for computer presentation was boring, repetitious, and dull--in other words, a perfect job for computers. The resulting solution took the form of programs that would generate items for students (e.g. Koffman & Blount, 1974) and was called Generative Computer-Assisted Instruction, although what we now mean by generative computer instruction is a little more sophisticated. In any event, it occurred to early observers of the scene that since we were trying to use computers to mimic the item generation capabilities of expert human tutors, why not use computers to mimic all the capabilities of human tutors? Thus was born the notion of computerized tutorial dialogue.

The development of computerized tutorial dialogues involves the application of artificial intelligence techniques to computer instruction, resulting in the information structure oriented (ISO) approaches discussed and advocated by Carbonell (1970). Carbonell contrasted these approaches with ad hoc, frame oriented (AFO) approaches based on techniques of programmed instruction. Carbonnell pointed out that, unlike AFO approaches, ISO approaches can be used to develop instructional systems that answer questions not specifically anticipated by the instruction designers, construct appropriate questions on given topics, and carry on a "mixed-initiative" dialogue in which either the student or the computer can introduce a response, topic, or idea in a free and comfortable subset of English. This may sound like programming a computer to be an expert tutor, and it is meant to.

This approach is in the mainstream of current developments in cognitive psychology which have taught us--or reminded us--that perception and learning are overwhelmingly constructive processes (cf. Resnick, 1983). In perception we do not collect bits of information from the "outside world" and paste them up on perceptual templates, and in instruction we are not writing information on blank slates in students' heads. Instead, we are dealing with active and very rich simulations of the world which students must create in order to perceive or learn. It is analysis by synthesis with a vengeance, and what gets transmitted in communication and instruction are not bits of information or knowledge but cues that may or may not be used to adjust the simulations being built up by students. The attempt in tutorial dialogue approaches is to deal directly with these simulations in ways that no drill and practice program--and no book--can.

Computers are both very good at this and very bad. Consider the following sentence:

The man the dog the girl owned bit died.

This is a difficult sentence for us to parse. We quickly become entangled in its syntactic nestings. Human chauvinism leads us to assume that since the sentence is difficult for us to parse, it is impossible for a machine. Yet a computer could quickly discern, after diving into its recursive routines for processing nested constructions, that there was a dog that was owned by a girl, that the dog bit a man, and that the man subsequently died.

Here is another example:

The man kicked the ball kicked the ball.

This is a perfectly grammatical sentence, as any self-respecting machine would discover after reversing an English transformational rule for deleting function words and determining that a man to whom a ball was kicked, kicked the ball back. In both these examples, a computer is less likely than we are to be confused or distracted, and its ability to process these two examples illustrates real intellectual ability. "Artificial intelligence" is, after all, a poor name for the business of making computers intelligent. Intelligence, or intellectual ability, is really what the field is all about. That theories of intelligence are tested by algorithmization and putting them on computers is merely an issue of methodology, albeit a central one; there is nothing artificial about the capabilities targeted by this work.

Next we might consider the following example, taken from Donald Norman (1973):

What was Charles Dickens's telephone number?

A knowledgeable program would search the attributes it had associated with Charles Dickens looking for a telephone number, and/or it would search its list of telephone numbers looking for one associated with Charles Dickens. Presumably, no telephone number will be found for Charles Dickens, and this fact will simply be reported. As most human information processors must be aware, there is a more intelligent answer to this query that completely avoids searching lists of telephone numbers and/or attributes. To reply that telephones were not used until after Dickens's lifetime does require some knowledge, but this information could easily be stored by the computer. The real problem is to devise a computer program that knows when this information is relevant.

These examples also illustrate the distinction drawn by John Brown between "intelligent" computer instruction and what might be called "knowledgeable" computer instruction. In this light we might consider the following student/computer protocol taken from Brown's SOPHIE system (Brown, Burton, & Bell, 1974). In this example, the student is completing a series of tests on a computer simulated piece of electronic equipment (a regulated power supply) with a fault generated by the computer. Student reponses are underlined.

<u>WHAT IS THE BSE EMITTER VOLTAGE OF THE VOLTAGE LIMITING TRANSITOR?</u>

BASE

TRANSISTOR

THE BASE EMITTER VOLTAGE OF Q5 IS -.037 MILLIVOLTS.

The spelling corrector has detected and corrected two misspellings. The system must also know that Q5 is the voltage limiting transistor.

<u>WHAT ABOUT THE CURRENT LIMITING TRANSISTOR</u>

THE BASE EMITTER VOLTAGE OF Q6 IS .349 VOLTS.

The meaning of this question is completely dependent on the context established by the prior sentence. The system uses this context to determine that the student is interested not only in the voltage measurements but in the voltage across a particular junction.

WHAT SHOULD IT BE?

IN A WORKING CIRCUIT THE BASE EMITTER VOLTAGE OF Q6 IS .593 VOLTS.

Again the context determines the meaning of the question. Moreover, the system must infer that "should" implies what the measurement would be in a properly working device.

REPLACE Q6

I AM GOING TO ASK YOU SOME QUESTIONS ABOUT HOW Q6 IS FAULTED. ARE ANY JUNCTIONS SHORTED?

At this point SOPHIE knows that the student is barking up the wrong tree. Rather than simply replacing the transistor as requested, SOPHIE begins to query the student about what he or she thinks is wrong. If the student gets at least something right, the system says what was actually wrong and goes ahead and replaces the faulty part.

Clearly, SOPHIE has progressed beyond knowledgeable syntactic and semantic processing of student inquiries and is behaving in an "intelligent" manner. The system tries to create a reactive environment in which the student learns by actively trying out ideas in interaction with a simulation program. However, the program does more than simply simulate the information to be transmitted; it provides for tutorial feedback and, in effect, for a one-to-one relationship with an "articulate expert" problem solver who helps the student create, experiment with, and debug his or her own ideas.

Several reviews of this area have appeared, notable among which are discussions by Peele and Riseman (1975), Sleeman and Brown (1982), Barr and Feigenbaum (1982), and Fletcher (1984). Fletcher references about 16 of these tutorial dialogue systems that have been or are being developed. Carbonell's SCHOLAR (1970) and Brown's SOPHIE (Brown, Burton, & Bell, 1974) were seminal systems in the development of tutorial dialogues. The two premier systems currently seem to be GUIDON (Clancey, 1979) and Steamer (Williams, Holland, & Stevens, 1981).

GUIDON serves as a physician's consultant for the student, who plays the role of the physician, in diagnosing infectious diseases. GUIDON focuses directly on the problems a subject matter expert faces in making his or her expertise, understanding, and heuristics accessible to students. GUIDON takes account of students' knowledge and interests in choosing what to present, it incorporates a knowledge base that is augmented to better organize and explain the subject matter to the student, and its teaching expertise is represented explicitly and modularly so that it can be modified for different research designs. GUIDON both "knows" the subject matter and can explain to the student the paths it uses to reach a diagnosis just as an expert tutor does.

Steamer is a computer-based system being developed by the Navy to provide instruction in steam propulsion engineering. It links a very complicated and highly abstract, quantitative (mathematical) model of a ship's steam propulsion system to high quality visual (graphics) presentations of the underlying model. The student is thereby able to manipulate the underlying abstract model through the graphics interface and to see in computer graphics presentations how the effects of these manipulations would be propagated throughout the ship's steam propulsion system. Additionally, Steamer uses the student's manipulation to better model his or her understanding of steam and to extend, correct, and deepen that understanding.

At this point, we may all wonder if we are going to see tutorial dialogue systems of this sort in our classrooms in the near future. About a year ago one of the major figures in the tutorial dialogue world passed through Oregon State University leaving the following quote in his wake: "It's amazing what you can do when you only have two megabytes of memory."

To those of us used to working with 32K and 64K byte personal computers, the notion of 128K bytes seems like Nirvana. Two million bytes is beyond all imagining, and this is apparently the low end for someone working with tutorial dialogues. The point is that the computational requirements for tutorial dialogue systems are very large. A single user system sufficiently powerful for delivery but not development of tutorial dialgues might be purchased today for about $20,000. In ten years the picture will change completely, and for this reason the development of tutorial dialogue systems should now be pursued vigorously on large machines.

Somewhere among all the new directions for computer courseware a major breakthrough will occur. Tutorial dialogues appear to be a likely area for this breakthrough. This direction represents an approach that is both evolutionary and revolutionary. That is to say, we can expect it to help us accomplish what we want to do now and to alter in very fundamental ways our understanding of what instruction should be. In any event, tutorial dialogues could not be implemented without computers, and their development is limited by the current state of the art in both computer hardware and software. It is often said that hardware and software developments are far in advance of our capabilities to use them in instruction. In the case of tutorial dialogues, this is not true. We are simultaneously developing and capitalizing on the state of the art in computer hardware and software technology.

Much still needs to be done. We need to learn how to represent imperfectly understood and poorly described knowledge domains and to reduce the costs of creating knowledge domains. Better natural language processing must be developed, techniques for modeling learners must become far more sophisticated, and our understanding of what master tutors and teachers do must be greatly enhanced. We need to learn how to interface computer tutorial dialogues with the practice of classroom teachers. However, these issues only indicate that breakthroughs in this area will occur perhaps later rather than sooner. The promise of tutorial dialogues for improving instruction remains.

This promise is particularly evident when we review efforts to join tutorial dialogue techniques with simulation, the topic of the next section. In fact, we have already skirted these shoals very closely. After all, the student troubleshoots a simulated power supply in SOPHIE, diagnoses an ailing simulated patient in GUIDON, and operates a simulated steam propulsion system in Steamer. It may be past time to turn to the area of simulation in instruction.

Simulation

The currently strong and growing interest in simulation used for education is far overshadowed by the interest in and support for simulation used in training, specifically military and industrial training. Most readers will be familar with the long history and use of multi-million dollar aircraft simulators--some costing more than the aircraft they simulate--by the military and by aircraft manufacturers for pilot training. Twenty years ago, if one mentioned the use of simulators in instruction the reference would be to aircraft simulators and probably nothing else. The advent of computer technology has permanently altered this state of affairs.

Because current simulators are based on programmable computers, they need not be single purpose, representing only a single system such as the cockpit of an F-14 fighter aircraft. Instead, a wide range of related systems can be simulated for the purposes of training individuals who must learn to operate and maintain them. The Navy's Generalized Maintenance Trainer/Simulator (GMTS) (Rigney, Towne, King, & Moran, 1978) is a case in point. The GMTS can be used to simulate any device in which signal paths and their relationships to controls, indicators, and test points can be defined. So far the GMTS has demonstrated its versatility by being used to teach techniques to maintain both a radar repeater and a UHF communications systems.

Again because current simulators are based on programmable computers, they can be much smaller and less expensive than they were originally. Simulators too are benefitting from the micro-electronic revolution. The idea of "suitcase simulators" abounds in today's military. MITIPAC (Rigney & Towne, 1977), for instance, took the GMTS and shrunk it down via micro-electronics to fit into a suitcase-sized package which provides a true job site training capability. MITIPAC can now be transported to locations where military jobs are actually performed--in the field, on ships, on flight lines--and tailored to the specific jobs at hand. Many simulators have been built, tried, and evaluated in training, as Orlansky and String showed for training aircraft pilots (1977) and for training maintenance technicians (1981). In this sense, simulation is an established and proven technique for instruction. However, development of simulation for instruction is far from finished. The field is particularly fortunate in that promising and dramatic new "functionalities" now exist. Three of these new functionalities are interactive movies, surrogate travel, and spatial data management. All three of these use computer-controlled videodiscs.

Interactive Movies: Interactive movies attempt to translate movie viewing into an active, participatory process. In effect, the viewer becomes the director and controls many features of the movie. Feature controls available to the viewer are the following:

1. Perspective. The movie can be seen from different directions. In effect, the viewer can "walk around" ongoing action in the movie or view it from above or below.

2. Detail. The viewer can "zoom in" to see selected, detailed aspects of the ongoing action or can "back off" to gain more perspective on the action and simultaneous activity elsewhere.

3. Level of instruction. In some cases, the ongoing action may be too rich in detail or it may include too much irrelevant detail. The viewer can hear or see more or less about the ongoing process by so instructing an interactive movie system.

4. Level of abstraction. In some instances the viewer may wish to see the process being described in an entirely different form. For example, the viewer might choose to see an animated line drawing of an engine's operation to get a clearer understanding of what is going on. In some cases, elements shown in the line drawings may be invisible in the ongoing action, e.g., electrons or force fields.

5. Speed. Viewers can see the ongoing action at a wide range of speeds, including reverse action and still frame.

6. Plot. Viewers can change the plot to see the results of different decisions made at selected times during the movie.

Surrogate Travel: Surrogate travel forms a new approach to locale familiarization and low cost instruction. In surrogate travel, images organized into video segments showing discontinuous motion along a large number of paths in an area are stored on videodisc. Under microprocessor control, the student accesses different sections of the videodisc, simulating movement over the selected path.

The student sees with photographic realism the area of interest, for instance, a city street or a hallway in a building. The student can then choose both the path and the speed of advance through the area using simple controls, usually a joystick. To go forward the student pushes forward on the joystick; to make a left turn the student pushes the joystick to the left; to go faster the student pushes the joystick harder, and so on.

The videodisc that frames the viewer sees originate as filmed views of what one would actually see in the area. To allow coverage of very large areas, the frames are taken at periodic intervals that may range from every foot inside a building, to every ten feet down a city street, to hundreds of feet in a large open area, e.g., a harbor. Coverage of very small areas is also of interest. In microtravel, which is a combination of surrogate travel and interactive movies, travel is possible where humans could could never go: inside watches while they are running, inside living organisms, etc.

The rate of frame playback, which is the number of times each video frame is displayed before the next frame is shown, determines the apparent speed of travel. Free choice in what routes may be taken is obtained by filming all possible paths in the area as well as all possible turns through all intersections. To some extent this is a time-consuming and expensive technology, but it has become relatively efficient because of the design of special equipment and procedures for doing the filming.

Demonstrations of this technology have been developed for building interiors (National Gallery of Art), a small town (Aspen, Colorado), an industrial facility (nuclear power plant), and San Francisco Harbor. Plans are underway to produce a prototype video map library of broader scope for selected areas worldwide.

Spatial Data Management: Basically, spatial data storage and retrieval of information is the method of loci transformed to a video or computer graphics format. The information is stored and retrieved through its association with already familiar geographic terrain.

Suppose, for instance, a student wanted to study the musical environment in which Ralph Vaughan Williams wrote his "Concerto for Tuba and Orchestra". In an ordinary data retrieval system the student will type in a complicated set of Boolean expressions--or English phrases standing for Boolean expressions--and will receive in return only textual information about the topic. Relevant information closely related to the information successfully retrieved will not appear unless the student starts from the top again with a new set of Boolean expressions. In a spatially organized data system, the underlying geography will be familiar to the student, for instance the school campus. The student may then "fly" to the music department (or library, concert hall, professor's office, etc.) and look for a tuba (or an orchestra, music library, portrait of the composer, etc.). Upon finding a tuba or other relevant cue, the student can "zoom" into it, still using his single joystick contol, select the concerto by name (or by hearing it, seeing the score, seeing the composer, etc.) and then hear, see, and read more information about it all retrieved through visually oriented associations.

In this way, spatial data management acts as an electronic library that gives students and instructors access to a wide assortment of multi-source and multi-media information whose components are associated in a natural and easily accessible manner. Instructors can access the system to create and/or assemble their own information spaces to be explored later by their students or subsequently present these materials to large audiences in single locations using large screen television projection or to multiple locations though cable distribution systems. Students can independently use the system for individualized instruction by working though previously designed information spaces, by browsing on their own, or by creating their own data spaces. When students and instructors are in remote locations, offsite instruction can be facilitated by linking two or more systems together using regular telephone lines. In this manner, a student or instructor can "fly" the other to a topic of interest, sharing at geographically remote sites a large, visually-oriented library of information.

Two points are worth noting about these new directions for simulation applied to instruction. First, they cannot be implemented in a book. Second, the application of these new directions for simulation-based computer instruction in education is just beginning. One can easily imagine application of this technology to science education. Perhaps a few words on this subject are in order.

The best way to learn science is by doing it. The excitement, mystery, frustrations, and triumphs of science are only dimly revealed by the usual fare of introductory science course. It would be far better for students, especially introductory students, to approach science with freedom to indulge their curiosity, form and re-form their own hypotheses, design and perform their own experiments, and build their own models and theories to explain natural phenomena. Unless there are drastic shifts in national funding policies for science education, this essential scientific experience will be prohibitively expensive to provide. The result is that students--especially elementary and junior high school students--are "turned off" by science at a time when our industrial and academic need for scientists, engineers, and technologists is acute and increasing.

What is needed in science education is something that has the impact of video gaming, but at the same time possesses substantial pedagogical power. One way to accomplish this is to provide simulated scientific experiences to students. Good simulations are exciting, compelling, and teach effectively by providing an environment in which learners must live with their decisions. Simulated experiences need not replace existing laboratory and field exercises, but they may expand and supplement them. Moreover, simulated experiences may be superior to real experiences in at least four ways. First, and primarily, simulation can be economical. Use of simulation should reduce the need for laboratory equipment and its maintenance, laboratory supplies, and travel costs for field experience. Second, simulation can make relevant phenomena more readily visible in two ways. In one way it can make the invisible visible. For instance, the flow of ions can be seen more clearly and simply under simulated conditions than under real conditions. In another way, simulation may increase the visibility of a phenomenon by separating it from a confusing or chaotic background. One can see the conceptual forest without getting lost in the procedural trees. Third, simulation allows reproducibility. Students can replay over and over chains of events that they could not otherwise observe repeatedly. Fourth, simulated experience is often safer than the real thing. Airplanes can be crashed, poisons can be ingested, and laboratories can be exploded with impunity in simulated environments.

Two sorts of relevant scientific experience that lend themselves readily to simulation are field study and laboratory experimentation. These two kinds of experience could be provided using the new functionalities described above. These functionalities could be used to build video field trips and simulated laboratories.

In the field, the student sees the total ecological view. He/she sees the overall landscape, the terrain, the populations of organisms, and individual samples of interest in their special areas. In sciences such as biology, geology, paleontology, archeology, and even astronomy, substantial learning and appreciation can be achieved by travel to locations that are difficult to access under the best of conditions. However, field trips are treated as an instructional frill. After all, the trips are made rarely and locally (they depend for success on what is serendipitously nearby); they emphasize only the group (individuals do not have an opportunity to do the science on their own); and most of the administrative effort centers on getting to the field and getting back, not on the field experience itself. As a result, even short, local field trips are being cancelled by schools because their cost in time and fuel is not balanced by their educational return. Surrogate travel removes the major objections to field experience and offers to each student a broadened opportunity to experience scientific phenomena in their natural, ecological context.

Students interested in, say, the biology of deserts could visit the Gobi in the morning, the Sahara around noon, and the Sonoran in the afternoon. They could travel around in each habitat locating, identifying, and "gathering" samples in roughly the same way, and for the same purposes, as a trained scientist. Panning and zooming through the full range of habitats could develop in students many of the same intuitions and understandings of environmental, geographic, and climatic contexts that an experienced scientist gains from actual travel.

Back in school, laboratories provide a problem solving environment where students interact, observe processes, and are stimulated to synthesize concepts as part of their learning. However, many schools are eliminating laboratories from their science courses, not because they are not useful learning experiences, but because of the cost of obtaining, maintaining, and supporting specimens, samples, and laboratory equipment. Interactive movies and spatial data management allow us to simulate laboratory experiences without the high cost and effort that is normally involved under the present pattern.

Students can create, store, and retrieve information from mammoth data banks using spatial data management. One can imagine high school students organizing an entire archaeological excavation or geological survey using spatial data techniques. One can also imagine elementary school students setting up and running high-energy particle physics experiments through interactive movies with plot control. Students would also have full use of the latest in telescopes, microscopes, and even endoscopes through computer-based simulation.

Finally, laboratory and field experiences can be linked so that hypotheses developed in the laboratory would be tested by return "travel" to the correct habitat, "collection" of data or specimens, and return to the laboratory for testing and verification. In this way, the excitement, frustrations, and triumphs of scientific experiences would become accessible to students.

In the above, simulation was presented as a new direction that is finding its way into computer instruction, but it is interesting to note that the history of computer instruction is exactly the reverse.

The first use of computers to teach grew out of a computer-based system that was primarily intended for simulation of real world experiences. This was the Air Force's SAGE (Semi-Automatic Ground Environment) system which was built in the late 1950's to train Air Force personnel in the techniques and tactics of air defense (Rowell & Streich, 1964; Parsons, 1972). Computers in SAGE were inititally used to simulate equipment, mostly radar, to which ground based air defense personnel were to make appropriate reactions. However, as time progressed, the SAGE computers began to be used to present training in a more general-purpose fashion.

The University of Illinois's PLATO (Programmed Logic for Automatic Teaching Operations) was probably the first computer system built specifically for computer instruction. Interestingly, it too was first supported solely by the military--in this case by the Army Signal Corps, the Office of Naval Research, and the Air Force Office of Scientific Research (Bitzer, Braunfeld, & Lichtenberger, 1962). Initially PLATO was used as a sort of "book with feedback" following the suggestion of Chalmers Sherwin, and few who saw early demonstrations of PLATO in the late 1960's were able to escape its "fruit fly" demonstration. This was a simulated biology laboratory showing in high quality graphics successive generations of fruit flies as they illustrated a model of genetics. This type of simulation in computer instruction is still in use.

The focus in this section is on new techniques for simulation, three of which are listed above. These three have been discussed in a little more detail by Bolt (1979) and by Levin and Fletcher (1981).

Other techniques may well be on the way. We have barely begun to explore the instructional possibilities of natural language processing (as opposed to computer language processing), voice output, voice input, computer-generated imagery (which may obviate some of the need for videodisc storage), and psychoneurological monitoring. New functionalities for these capabilities will doubtless be developed. However, it should be emphasized that this process of discovery is at least as demanding of time, resources, and ingenuity as the development of the computational capabilities themselves. Swamping schools with hardware and computer capabilities and then expecting instructional functionalities to flow spontaneously in their wake is simply wrong. The process will continue to require support, encouragement, resources, and time.

FINAL WORD

It is wrong to inundate our educational institutions with new technologies without insisting that they do at least something to help us through the day. It is also wrong to hold off all investment in new technologies because they may affect what it is we want to do. The correct approach seems to be somewhere in the middle. No one envisioned teleconferencing when the telephone was invented, no one imagined our current interstate highway transportation system when the horseless carriage came along, and steam engines languished for 30 years pumping water out of coal mines before someone began to think seriously of their possibilities for self-locomotion. We have benefitted from the introduction of these devices into our lives just as we have suffered from them. We must give the new technologies their place if we are to improve our instructional practice as the Gardner Commission said we must. At least in the case of computers, we are in a position to insist that they be of some immediate practical value along the way. This is a fortunate position to be in, and we should capitalize on it. Computers can help meet goals and solve current problems of schools and school districts at the same time that they are helping to advance the craft of instruction. We can and should expect them to do both.

In short, computers will help us better perform the business of instruction as we envision it today. They will also broaden our horizons. They will change and expand our ideas about what instruction is and what it must do. Their challenge to us as educators is as serious as their promise. We should rise to the occasion.

REFERENCES

Atkinson, R.C., & Paulson, J.A. An approach to the psychology of instruction. Psychological Bulletin, 1972, 78, 49-61.

Barr, A., & Feigenbaum, E.A. Handbook of artificial intelligence: Vol. 2. Los Altos, CA: William Kaufman, 1982.

Bitzer, D.L., Braunfeld, P.G., & Lichtenberger, W.W. PLATO II: A multiple-student, computer-controlled, automatic teaching device. In J.E. Coulson (Ed.), Programmed learning and computer-based instruction. New York: John Wiley and Sons, 1962.

Bolt, R.A. Spatial data management. Cambridge, MA: Massachusetts Institute of Technology, 1979.

Brown, J.S., Burton, R.R., & Bell, A.B. SOPHIE: A sophisticated instructional environment for teaching electronic troubleshooting (An example of AI in CAI) (BBN Report 2790). Cambridge, MA: Bolt Beranek and Newman, Inc., 1974.

Bunderson, C.V. Courseware. In H.F. O'Neil, Jr. (Ed.), Computer-based instruction: A state-of-the-art assessment. New York: Academic Press, 1981.

Carbonell, J.R. AI in CAI: An artificial intelligence approach to computer-assisted instruction. IEEE Transactions on Man-Machine Systems, 11, 190-202, 1970.

Chant, V.G., & Atkinson, R.C. Optimal allocation of instructional effort to interrelated learning strands. Journal of Mathematical Psychology, 1973, 10, 1-25.

Clancey, W.J. Tutoring rules for guiding a case method dialogue. International Journal of Man Machine Studies, 1979, 11, 25-49.

Crowder, N.A. Automated tutoring by means of intrinsic programming. In E.H. Galanter (Ed.), Automatic teaching: The state of the art. New York: John Wiley and Sons, 1959.

Fletcher, J.D. Models of the learner in computer-assisted instruction. Journal of Computer-Based Instruction, 1975, 1, 118-126.

Fletcher, J.D. Computer-assisted instruction in beginning reading: The Stanford projects. In L.B. Resnick & P.A. Weaver (Eds.), Theory and Practice of Early Reading: Vol. 2. Hillsdale, NJ: Lawrence Erlbaum Associates, 1979.

Fletcher, J.D. Intelligent instructional systems in training. In S.A. Andriole (Ed.), Applications in artificial intelligence. New York: Petrocelli Books, in press.

Fletcher, J.D., & Beard, M.H. *Computer-assisted instruction in language arts for hearing-impaired students* (Technical Report No. 215). Stanford, CA: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1973.

Gilbert, T.F. On the relevance of laboratory investigation of learning to self-instructional programming. In A.A. Lumsdaine & R. Glaser (Eds.), *Teaching machines and programmed learning: A source book*. Washington, DC: National Educational Association, 1960.

Koffman, E.B., & Blount, S.E. A modular system of generative CAI in machine-language programming. *IEEE Transactions on Systems. Man, and Cybernetics*, 1974, *4*, 335-343.

Laubsch, J.H. Optimal item allocation in computer-assisted instruction. *IAG Journal*, 1970, *3*, 295-311.

Levin, S., & Fletcher, J.D. Videodisc technology: An approach to the design of training devices. In J. Moraal & K.F. Kraiss (Eds.), *Manned systems design: Methods, equipment, and applications*. New York: Plenum Press, 1981.

Lorton, P.V. *Computer-based instruction in spelling: An investigation of optimal strategies for presenting instructional material*. Doctoral dissertation, Stanford University, Stanford, California, 1973.

McLuhan, H.M., & Fiore, Q. *The medium is the message*. New York: Random House, 1967.

Orlansky, J. & String, J. *Cost-effectiveness of maintenance simulators for military training* (IDA Paper P-1568). Arlington, VA: Institute for Defense Analyses, 1981.

Papert, S. *Mindstorms*. New York: Basic Books, 1980.

Parsons, H.M. *Man-machine system experiments*. Baltimore, MD: Johns Hopkins Press, 1972.

Paulson, J.A. *An evaluation of instructional strategies in a simple learning situation*. Doctoral dissertation and Technical Report No. 209. Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, California, 1973.

Peelle, H.A., & Riseman, E.M. Four faces of HAL: A framework for using artificial intelligence techniques in computer-assisted instruction. *IEEE Transactions on Systems, Man, and Cybernetics*, 1975, 5, 375-380.

Resnick, L.B. Mathematics and science learning: A new conception. *Science*, 1983, *220*, 477-478.

Rigney, J.W., & Towne, D.M. Some concepts, devices, and procedures for improving technical training and on-the-job performance of maintenance (Technical Report 76-C-0023-1). Orlando, FL: Naval Training Equipment Center, 1977.

Rigney, J.W., Towne, D.M., King, C.A., & Moran, P.J. Field evaluation of the generalized maintenance trainer-simulator: I. Fleet communications system (Technical Report No. 89). Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1978.

Rowell, J.T., & Streich, E.R. The SAGE system training program for the Air Defense Command. Human Factors, 1964, 6, 537-548.

Rumelhart, D.E. The effects of interpresentation intervals on performance in a continuous paired-associate task (Technical Report No. 116). Stanford, CA: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1967.

Sleeman, D., & Brown, J.S. (Eds.). Intelligent tutoring systems. New York: Academic Press, 1982.

Suppes, P. Problems of optimization in learning a list of simple items. In M.W. Shelley & G.L. Bryan (Eds.), Human judgments and optimality. New York: John Wiley and Sons, 1964.

Suppes, P. Some theoretical models for mathematics learning. Journal of Research and Development in Education, 1967, 1, 5-22.

Suppes, P., Fletcher, J.D., & Zanotti, N. Performance models of American Indian students on computer-assisted instruction in elementary mathematics. Instructional Science, 1975, 4, 303-313.

Suppes, P., Fletcher, J.D., & Zanotti, N. Models of individual trajectories in computer-assisted instruction for deaf students. Journal of Educational Psychology, 1976, 68, 117-127.

Williams, M., Hollan, J., & Stevens, A. An overview of STEAMER: An advanced computer-assisted instruction system for propulsion engineering. Behavior Research Methods and Instrumentation, 1981, 13, 85-90.