Design of instructional dialogs to be delivered by computer

J. Nievergelt
University of Wollongong

Follow this and additional works at: https://ro.uow.edu.au/compsiwp

Recommended Citation
Design of Instructional Dialogs to be Delivered by Computer
========================================================

J. Nievergelt
Informatik, ETH, CH-8092 Zurich, Switzerland
(This work was done while the author was visiting the University of Wollongong, NSW 2500, Australia)

ABSTRACT
=======

This aims to be a pragmatic introduction to Computer-Assisted Instruction (CAI) for people who consider introducing CAI as part of their instructional operation. Since the set-up costs (in particular, man-power) for CAI are considerable, a careful analysis of expected costs and benefits is a must in order to prevent later disappointment. We list some of the main questions that should be raised and answered, and summarize the collected know-how and experience from a number of CAI projects. This survey covers a brief history of the development of CAI, requirements on the environment (hardware, software, administrative, personnel), and a manual of style for writing instructional material to be delivered by computer.
1. Introduction

2. A brief history of CAI
   1. Ideas that shaped early CAI
   2. Reassessing the situation
   3. Computers, problem solving, and general education
   4. The diversity of current CAI projects
   5. What does this all have to do with education?

3. What you need to consider before you get started
   1. Hardware
   2. System software
   3. Author language
   4. Courseware
   5. Instructional and administrative facilities
   6. Personnel
   7. Costs

4. Design of instructional dialogs: strategic considerations
   1. Choice of topic
   2. What, to whom, and how?
   3. Setting the level of ambition: smart or dumb tutor?
   4. The tutor should not pretend to be smarter than it is
   5. Coercion vs. laissez-faire, or: the student is smarter than your program
   6. Tone of conversation: a picture (and other things) is worth a thousand words

5. Design of instructional dialogs: manual of style
   1. Know your medium: the computer-driven screen
   2. What we can learn from other media
   3. Educational hang-ups
      1. on control and reinforcing
      2. don’t collect more data than you are willing to look at
   4. Programmer’s hang-ups
      1. clock control is fun, but user control is better
      2. I know you can parse it, but there should be no need to
      3. defaults and options
   5. The dynamic page: don’t scroll or scramble
   6. Structure, maps, and (fast) motion

6. Concluding thoughts

7. Bibliography
1. Introduction

Computer-assisted instruction (CAI) has experienced two turbulent decades of development since its beginnings in the late fifties. High hopes of having found a labor saving technology of education have been alternating with disillusionment at the unimaginative use of computers for electronic page turning. The record of actual use of CAI has also been spotty. Some large scale development efforts have yielded only a small pay-off to date; it is anybody's guess whether the CAI products currently on the market will turn out to be a commercial success. On the other hand, the recent proliferation of "smart" machines (for example terminals, typewriters, cash registers) containing a microcomputer, keyboard and screen, and hence capable of conducting an instructional dialog with a user, has opened a wide new area of application of CAI - the realization of the self-explanatory machine, which permits a casual user to learn to operate this machine by interacting with it. I believe that these ad hoc uses of CAI will ultimately succeed where frontal attacks have failed: to introduce the computer-delivered instructional dialog as a standard tool into education. In sections 2 and 6 some arguments are given in support of this prediction.

Regardless of the long range outlook for CAI, a new situation has arisen in the last few years. Because the hardware needed to deliver instructional dialogs is now available on many systems which were designed for other purposes, many more people than ever before find themselves faced with the decision of whether or not to use CAI in some limited application; and if so, how to approach this unfamiliar task.

The literature on CAI is vast but repetitive. This paper attempts to list the main questions that must be raised and answered before CAI is introduced; and to summarize the collective know-how and experience of a number of CAI projects with which I have been involved, or have had the opportunity to observe at first hand.

CAI is a field where experience and common sense are the only guidelines - there is no relevant theory to guide the designer, administrator, or user. This insight came to the CAI community relatively late - after a decade of domination by educational and psychological theories. The main lesson the CAI practitioner can learn today from these theories fashionable during the sixties is to avoid their major mistake - which was to impose a straightjacket on an emerging field (restriction to a few rigidly defined teaching strategies), before a sufficient number of alternatives had been explored. While the current lack of theories is understandable, it makes it difficult to say anything conclusive in this field. Almost any assessment of the computer's abilities as a medium for delivering instruction is based on personal experience, in a particular environment, and can often be disputed by experimenters who have tried a similar approach in a different environment. It is almost never possible to prove anything in a scientific sense, despite controlled experiments. I believe that the main thrust of this paper, and most recommenda-
tions in it, would be accepted by a majority of "users of CAI" - people from other disciplines who have experienced learning by means of CAI, as students or as instructors. Perhaps this paper would be rejected by some educational or psychological researchers in CAI, who tend to attribute to CAI a scientific status which, in my opinion, this field has not yet attained.

This paper is organized as follows. Section 2 presents a brief history of the development of CAI, in order to expose the reader to the spectrum of ideas that have shaped the current state of the art. You will find that practically any possible way to look at CAI has been prominent at some time in the past. Section 3 discusses the most important question for someone intent on entering the field of CAI: what do I need to get started, in terms of hardware, software, administrative facilities, and personnel? Sections 4 and 5 constitute a brief manual of style for authors of "courseware" - instructional dialogs to be delivered by computer. It is mostly common sense - which is usually violated by beginning authors, until it is pointed out to them by hundreds of complaints from students. A brief annotated bibliography provides entry points for the reader who wishes to make his own survey of the literature.
2. A brief history of CAI

Those who cannot remember the past are condemned to repeat it.
Santayana

2.1 Ideas that shaped early CAI

The intellectual environment that gave rise to the first generation of CAI systems in the early sixties was strongly influenced by the programmed instruction movement, which, in the view of its proponents, was based on a science of learning (e.g. Skinner [Sk 54]). CAI was seen as a direct continuation of the mechanical teaching devices (Pressey [Pr 26]), with the processing and decision-making ability of computers finally providing the flexibility whose lack had severely limited the use of mechanical realizations.

The dominant mood of optimism among workers in CAI was rationalized by arguments along the following lines:

1) education is a labor-intensive activity,
2) technology applied to other labor-intensive activities in the past has greatly increased productivity and cost-effectiveness,
3) with programmed instruction as a teaching strategy and computers as a delivery device, a technology of education has finally arrived, and hence
4) CAI will significantly improve education in the foreseeable future (make it more effective and cheaper).

The argument was sufficiently alluring to draw all kinds of scientific, technological, and commercial interests into the field of CAI. As prominent examples from this early phase of CAI let me mention the Stanford project, particularly the arithmetic drill-and-practice program (Suppes [SM 72]), and the IBM 1500 CAI system with the author language "Coursewriter", which became a model for most of the author languages provided by computer manufacturers. More information on CAI activities during the sixties can be found in a collection of articles by Atkinson and Wilson, [AW 69].

2.2 Reassessing the situation

Reality did not live up to expectations. By 1970 a number of facts and conclusions that dampened the early optimism and indicated that a reorientation was necessary were gaining acceptance:

1) CAI had not caught on as a means of routine instruction.
2) Programmed instruction and drill were not a universal technology of instruction; they had a rather limited domain of applicability.
3) Restriction to a few fixed teaching strategies, in particular those that impose rigid control of the dialog by the program, was unreasonable. Learning strategies where the user controls the dialog, such as inquiry and simulation, should be emphasized.

4) CAI was still significantly more expensive than conventional classroom instruction.

5) The goal of writing portable courseware in order to amortize the cost of lesson preparation among more users was not yet in sight.

6) The computer resources (terminal, processor, system software) required to implement an effective instructional dialog had been underestimated; the need for graphics and immediate response emerged as a necessity.

7) Resources had been diluted into too many projects of insufficient size; CAI research and development should be carried out by sizable groups of systems designers and authors.

Not everybody could be expected to agree with all of these points, but I believe they express fairly well the collective wisdom gained from experimenting with the first generation of CAI systems during the sixties. The consequence of this insight was that the CAI projects of the seventies showed a much greater variety of approaches, less dogma and more experimentation, than those of the first generation. Before discussing this new trend in section 2.4, I wish to touch upon a movement that started as a direct antagonist to CAI, and is partly responsible for opening a wider horizon of approaches to the problem of how the computer can assist instruction.

2.3 Computers, problem solving, and general education

During the mid-sixties, concurrently with the first generation of CAI projects and undoubtedly spurned by a missionary drive to enlighten the CAI enthusiasts, another movement to bring computers into education gained visibility. Its main premise was that conventional CAI exploited only a small part of the computer's power by restricting itself to "electronic page turning"; and that a computer is such a great tool and toy that its greatest educational impact will materialize only if students are given full control over it, that is, are programming it to solve problems of their own choice.

The most prominent representative of this movement is the LOGO project started by Feurzeig and Papert (see, e.g. [Pa 70]). An eloquent statement of the position that the main role of computers in instruction is as a subject to be taught rather than as a medium for presentation of instructional material was made by Luehrman [Lu 72] in a paper with the provocative title "Should the computer teach the student, or vice-versa?"

This rhetorical question implies a decision which, fortunately, need not be made. Today it is accepted that one need not make a sharp distinction between the use of computers as a device for delivery of instruction and as a tool for problem solving.
Dwyer’s SOLO project [Dw 71] has long combined a "dual mode" (the beginner interacts with a teaching program designed to guide him by the hand) with a "solo mode" (the more advanced student uses the computer on his own). When "teaching machine" and "problem solving tool" are viewed as dual mode and solo mode, respectively, it is clear that the distinction is one of degree, not of principle. The antagonism of the "problem solving" exponents towards CAI can only be understood historically, as a reaction against the trivial use of computers as "electronic page turners". The student should interact with the computer in whatever way is most pleasant, interesting, and conducive to learning: depending on the topic, the skill of the student, and other things, this can evidently include anything from drill to unsupervised programming.

2.4 The diversity of current CAI projects

If you don’t know where you are going, any road will take you there.

While there was widespread agreement in the early seventies that CAI had to undergo some major changes (see section 2.2) in order to succeed, there was considerable diversity of opinion as to the direction in which to go. At the risk of over-simplification, I present the following summary as being typical of the opinions held by people of different backgrounds.

Administrators:
- consolidate CAI research in a few large projects,
- develop portable CAI systems to enlarge the potential audience of courseware.

Educational experimenters:
- drop traditional CAI, teach the use of computers as problem solving tools.

Educational theoreticians:
- drop teaching strategies which enforce rigid program control, emphasize learner control.

Engineers:
- develop better hardware, in particular terminals.

Programmers:
- drop traditional CAI author languages with their built-in limitation to static frames and PI-type sequencing;
- move towards general-purpose high-level languages; if necessary, add facilities for interaction, e.g. graphic input/output and timing procedures.

All of these views are represented in the wide spectrum of CAI projects during the seventies, which may be called the second generation of CAI systems. In the seventies CAI in America has been dominated by two large projects which had considerable support from government agencies and industry: PLATO at the Univer-
sity of Illinois, later marketed by Control Data Corporation, and TICCIT at the MITRE Corporation. Both of these projects utilize hardware significantly more powerful than what was available during the sixties, particularly the graphics terminals. The two projects differ completely, however, in their attitude toward the preparation of instructional material.

While TICCIT is proud of the uniform style of its courseware, based on a theory of instruction and generated according to a systematic process (see, for example, Bunderson [Bu 72]), PLATO is equally proud of the "Darwinian approach" most of its authors take towards lesson writing: try everything you can think of, and if you keep your eyes open and are prepared to throw away unsuccessful material, the good stuff will survive. The latter approach leads to some excellent lessons along with a fair amount of poor material. Given that we are in an early state of development of the art of writing instructional dialogs, this situation is an unavoidable price we must pay for the education of authors.

The Learning Research Group at the Xerox Palo Alto Research Center has been developing the Smalltalk system designed to provide a powerful personal programming environment for "children of all ages" (see Kay [Ka 72]). It includes tools for painting and drawing, animation, music synthesis, storage and retrieval of document information, and other activities. Its aim is to show what today's and tomorrow's computer technology can contribute towards the realization of a powerful environment for problem solving in the style of Papert's LOGO project.

Many more CAI projects around the world could be mentioned in order to show the breadth of applications, goals, and points of view. There are production systems in routine use, often tailored to a particular audience. For example, a CAI system at the Rehabilitationszentrum Heidelberg, Germany, provides instruction for people with various impairments; all the courseware was written in APL. IBM's Field Instruction System offers CAI to maintenance personnel away from their home base, delivered by the equipment they are maintaining; an effective way to utilize waiting time that might otherwise be wasted. The US Armed Forces have a great variety of activities in CAI; [Fl 75] is a survey. Computer-Managed Instruction (CMI), where the goal is to guide and control the student's learning activities closely on an individual basis, regardless of what media are involved in these activities, is practiced in some training programs (see [SC 74] for a survey).

There are research projects that investigate the limits of application of artificial intelligence techniques in CAI. The resulting programs are usually called "tutors": each tutor encompasses a domain of discourse within which it can engage the student in a "free" dialog, often in natural language, where the questions being asked, and their sequence, are not explicitly predetermined by the author. They are deduced or constructed from a model that represents the body of knowledge within this domain of discourse (see, for example, Brown and Sleeman [BS 78]).
The most significant trend in CAI to emerge in the last few years is the proliferation of ad hoc CAI activities for fun and profit by people who do not consider CAI to be their major goal, but simply do it because it is convenient and possible on the equipment they happen to have. One need only visit a "computer faire" to see that all the hobby computers and tiny business computers offered display a collection of games, some of them educational, and often some programs for arithmetic or verbal skill practice advertised "to give your child the best education". As another example, it is common for smart terminals or microcomputers on the market to offer instruction tapes that explain the operation of the device to the novice user.

These developments show that CAI, although it has not yet had a large impact, is now in the public domain, accessible to anybody who has available an interactive system and does not fear the programming effort required to produce instructional dialogs. This state of affairs has never been true before - up to the mid 70's, one needed an expensive computer to do CAI. With the proliferation of inexpensive computers, and a corresponding increase in the number of people who know how to teach computers, it is becoming evident that a growing number of users are experimenting in teaching computers to teach students. This phenomenon may finally provide the mass market whose lack has severely discouraged investment in the production of courseware.

2.5 What does all this have to do with education?

Education makes good use of many tools and techniques, but none of these, except the computer as the latest arrival on the scene, have been dignified with a new name such as "computer-assisted instruction". Why don't we speak of blackboard-assisted instruction, book-assisted instruction, or writing-assisted instruction? (see Luehrman, [Lu 72]). Because we feel that the blackboard, the book, and even writing, useful as they are, are not really that crucial when it comes to distinguish different kinds of instruction. We feel that the teacher should plan his instructional strategy first on the basis of WHAT he must teach and TO WHOM, and only secondarily on the basis of what tools, in particular media, are at his disposal. We expect the competent teacher to know how to adapt his instructional strategy to any one of a dozen typical situations, such as classroom instruction using a blackboard or flip-chart or projector, or individual instruction when only the spoken word is available.

The name CAI puts an undue emphasis on the computer. We should simply view the computer-driven screen as another medium available to the teacher, which is capable of doing certain things well and others poorly or not at all. A fair amount of skill is needed to use this medium effectively, probably more than is required by other media. If the necessary skill and equipment are available, the teacher can decide intelligently in what cases the computer is an appropriate medium to deliver the instructional design he has selected on the basis of the more fundamental questions: WHAT and TO WHOM.
The question in the title of this section intends to raise a fundamental issue which everybody involved in CAI should answer to his own satisfaction. I suggest the answer: No more and no less than books, blackboards, films, and other media have to do with education. If so, we should perhaps drop the word CAI and thus stop giving undue emphasis to the computer over other media. As we speak of writing a book, preparing and delivering a speech, or producing a film, we may in the future, when the novelty of computers has worn off, simply speak of writing and using instructional dialogs.
3. What you need to consider before you get started

A lot more effort has been put into CAI than benefits have been reaped to date. Anyone who starts a CAI project must anticipate several years of development effort before the enhanced quality and productivity of his instruction begin to pay off. If the project is insufficiently planned, it is likely to remain a drain on resources forever. It is therefore well worthwhile to assess one's resources and environment carefully before embarking on what is certain to be a costly effort. Here are the major points to be considered.

3.1 Hardware

The terminal (user station, console, or whatever it may be called) is the hardware component of the man-machine interface. It is the only piece of hardware that the user tends to see or care about, and the success of a CAI application stands or falls with the quality of the terminal used. If this sounds like an exaggerated statement, consider the following thought. The CAI user usually has other sources from which he can study, such as books. If the terminal causes eye strain, if everything from Shakespeare to mathematical formulas is presented in upper case letters, if the screen is so small that natural units of presentation cannot be seen at one time, if an engineering or scientific topic is presented without pictures, if the response time to trivial commands such as NEXT exceeds a second, the rational user will decide that he can study the same material more efficiently elsewhere.

The following requirements are about minimal for a terminal to be used for CAI in a variety of subject areas.

Text

- 20 lines at 60 characters;
- Upper and lower case, common mathematical symbols
- all common punctuation signs, including accents
- needed in the language of instruction;
- Highly desirable: user defined characters.

Graphics

- Ability to draw lines quickly on a point raster of about 200 * 200 points;
- No flicker;
- Sufficient brightness to avoid eye strain.

A few subject matters require additional capabilities. Language instruction for beginners requires sound, biology requires pictures of much higher resolution than indicated above. The resolution quoted above suffices for simple line drawings as they are typically used in engineering and scientific textbooks, where the precise shape of an object is unimportant, but the connections between objects is crucial. To describe such connections in words is clumsy for author and reader, and a CAI system to be used for scientific subjects MUST have at least the limited kind of graphics described above. Its quality corresponds to that of
"blackboard graphics", and if teachers of science cannot do without a blackboard, there is no reason to expect that computers teaching science can do without the equivalent of a blackboard.

"Don't bother starting a CAI project unless you have graphics" is a hard lesson to swallow for the computer manufacturer who wants to offer an add-on CAI package to whatever system he is selling at the moment, and to the computer center manager who has a lot of alphanumeric terminals connected to a time-sharing system. Try the following test if you think this rule can be violated for your specific application: see whether the textbooks in your field use diagrams and pictures.

Next to graphics, fast response is a very important requirement. We discuss it in the next section since software rather than hardware is the usual bottleneck in this regard.

3.2 System software
---------------------

The commercially available system software on the market was developed under two premises which do not hold for CAI today, namely:

- processors are expensive, hence it is cost-effective to centralize all applications, no matter how diverse, onto one or a few large computers;
- the typical user does not need to work in a highly interactive mode.

The first assumption led to large operating systems whose primary purpose is efficient utilization of hardware, at the cost of responsiveness to user inputs. The scarcity of applications requiring highly interactive mode of use conspired with the first reason to let users tolerate response times in the range of a few seconds to a few minutes.

It is a physiological fact that events that occur within about one tenth of a second are perceived as being instantaneous. The computer community should accept the axiom that any interactive system must respond to trivial user requests "instantaneously", i.e. within 1/10-th of a second. Examples of trivial requests are inserting a character into a line, answering a multiple choice question, or asking for the next paragraph of text. Today's computer users have been conditioned to consider response times of a few seconds as "fast", but those who have worked on systems providing instantaneous response know that a one second response time slows down their work noticeably. The rational CAI user who feels his time is being wasted, and the continuity of his thoughts is being disturbed, will walk away from a slow system.

Subsecond response time to trivial requests is the most important requirement on system software. If it is not satisfied, write your own operating system or abandon CAI.
3.3 Author language

Almost all author languages available from manufacturers are intellectual children of the programmed instruction era (see section 2.1 "Ideas that shaped early CAI"). If all your courseware were of the PI-type, with static frames as output and character strings or numbers as input, any of these author languages would be a reasonable tool. But most courseware should contain some simple model of reality that the student can manipulate, and present a dynamic record (often in the form of animation) of the result of this simulation. The reason computer games such as "Moonlander" appeal to many users is precisely that simulation under user control is a form of dialog or teaching strategy which cannot be presented on any other medium (except toys, the oldest form of educational technology).

An author of CAI material must not be afraid of programming. Once he has learned to program, he will not be satisfied with the PI-type author languages on the market. He will prefer any conventional high level programming language which is suitable for writing interactive programs. This means that it must have a convenient set of I/O procedures for output onto the screen, input from keyboard and screen, and time control. Output must include simple line-oriented graphics. A data type "string" of variable length, with procedures for matching, cutting and catenating strings, is helpful. The ability to define and dynamically create processes that execute concurrently is particularly useful for animation and simulation. That's all you need for an author language.

3.4 Courseware

Perhaps in the future you will be able to dial up a library of courseware, transmit a program over the wire into your home computer, and thus you will have access to a large collection of instructional material at no effort and little cost to you.

Unfortunately, we are far from this ideal. Today it makes no sense to start a CAI project unless you yourself are willing to write the majority of the courseware you need. Even on a large CAI system such as PLATO, where thousands of hours worth of courseware exist, new user groups soon tend to write their own instructional material. Perhaps this is as it should be, since an instructor planning to use CAI should try his hand at authoring. Another reason for writing your own is that much existing courseware is not very good — for lack of experienced authors, lack of incentive for authors to go through the laborious process of validating their courseware with students, or lack of a sufficiently large audience of students. If and when CAI lessons will have as wide an audience as books have, this write-your-own syndrome will give way to a profession of authors of instructional dialogs. But if you plan for CAI today, you cannot wait for this change to occur.
For whatever reason, count on having to develop courseware. Estimates of author time needed to produce the equivalent of one hour's worth of student contact time range from 10 hours to over 100 hours. 10 hours is an exceptional performance which may be achieved by a very experienced author who knows exactly what he wants to say (because he has said it before). 100 hours is a realistic average over several authors and much courseware.

3.5 Instructional and administrative facilities

Assuming you could buy a perfect CAI system and just the right courseware for your purpose, it still would not help much if you just put it in a hallway, accessible to anybody who walks by. Schools have an administrative structure that determines responsibilities of teachers and students (work and studies to be completed) and rewards (salaries and diplomas). Unless the CAI system is integrated into this structure and has the support of administrators and teachers, it will not succeed. It may be hard to gain this support. Introducing a CAI system requires work on everybody's part to understand the system - how to operate it, what courseware exists and what must be written, what fraction of the instructional load it can assume, what its strengths and weaknesses are.

A CAI system, like any other reasonably complex machine, requires a human organization around it to be effective.

3.6 Personnel

One of the early justifications for CAI was to overcome the shortage of teachers. Perhaps this may become true at some future time, but in the mean time the shortage of teachers has turned in an oversupply in most fields, and experience with CAI has rarely shown that teachers are being replaced by computers.

One must distinguish two kinds of students. The exceptional one is highly motivated and needs nobody to organize his studies for him - he will find sources from which he can learn, and if a CAI system is available to him, he will profit from it without human supervision. The typical student needs the psychological reassurance ("hand-holding") that only a person can provide. If this is lacking, he is likely to lose interest and relax in his efforts. It may be ideal for a teacher to be relieved of lecturing duties (which are relegated to the computer) and thus have more time to counsel students individually. In this sense CAI may greatly contribute to improving the quality of education. What it means, however, is that CAI is unlikely to replace a significant number of teachers, and will certainly introduce some authors of courseware.

Don't try to justify CAI as a labor saving technology - at least not yet. It must be justifiable as an improvement in the quality of education.
3.7 Costs

Cost estimates of CAI usage have varied by over a factor of 10, from a fraction of a dollar per student contact hour, to about 10 dollars (charged by some commercial establishments). There is no reason for CAI to be more expensive, in the long run, than television. The processing power that must be added to a TV set to turn it into small computer and screen capable of delivering instructional dialogs is getting cheaper all the time; its cost may settle between 100 and 1000 dollars. The broadband communications channels of TV are not needed - a telephone wire to a central repository of CAI material, or even the mail, are adequate to transmit programs to the user's computer, where they are executed. The preparation of CAI courseware need certainly not be more expensive than that of TV programs, so if their costs can be amortized over a large audience, education could be paid for entirely by advertisement, as TV is in some countries.

But until CAI has become a mass consumption item, it remains expensive. Certainly more expensive than a teacher facing a class of 20 students in the traditional school setting, even at the university level. And remember that CAI tends to become an add-on cost, without significant savings in reduced personnel.

CAI is already cost competitive in certain industrial training situations, where the "students'" salaries during instruction time are by far the dominant cost. If CAI shortens the training time by just 10%, it may be cost effective. I have observed such an operation in the training of airline flight crews (in the operation of new planes, in learning about airports unfamiliar to them, or new regulations), who are available in small groups at odd times of day or night, for periods of varying duration. CAI allows each crew member to study anything he wants at anytime he wants; traditional classroom instruction would imply reserving certain days for training, and making the crews unavailable for work.

If you run an expensive, specialized training operation now, chances are that CAI may be cost-effective. If yours is a run-of-the-mill school or university, CAI cannot be justified on the basis of current savings. It may be justified as an experiment or investment in the long range improvement of education.
4. Design of instructional dialogs: strategic considerations

An ounce of prevention is worth a pound of cure.

Writing a CAI lesson is similar to planning any other act of communication. First, certain global or strategic decisions must be made: WHAT is to be communicated, to WHOM (what background does he have), HOW (what sequence of ideas are to be presented, actions to be solicited). In this respect planning for CAI differs little from planning a speech, an article, a film, or a personal tutoring session. The author knows, however, that his medium has certain fundamental possibilities and limitations, and he rejects early those approaches that violate these limitations, or those that fall far short of the inherent possibilities. This section discusses the possibilities and limitations peculiar to computers, and how they affect the strategic decisions in designing CAI material.

Our discussion excludes specialised simulations (e.g. flight simulators for pilot training). These are not fundamentally different from less demanding simulations, such as "Moonlander", that are routinely played on hobby computers. But their hardware requirements are so extensive that they cannot be realistically implemented on today's CAI systems.

4.1 Choice of topic

Some attributes that apply to intellectual topics are: difficult or easy, routine or demanding creativity, requiring memorization or understanding, objective or subjective, and others. The point of thus categorizing topics is to find an efficient match between topic, student, and method of presentation. In CAI the population of possible students appears to be unrestricted (all age levels from kindergarten to professionals have been tried), while the possible methods of presentation are strongly restricted. It is impossible, for example, for a CAI program to "sit at the other end of a log", or to lead a Socratic dialog, where the student must assume as much initiative as he can, while the tutor only provides gentle guidance. If you try this, you will find that the computer tutor, like big brother, will provide a rigid guidance. It isn't smart enough to be gentle.

The CAI author's first question is to determine what kinds of topics lend themselves to the "rigid" type of presentation that a program can provide. Rigid here does not necessarily mean that it forces the student to follow a unique path. It may be as liberal as a library, which lets the user pick any item he wants out of its collection. Rigid in this context means that whenever the student brings up any idea outside the limited domain of discourse of the program, he meets with a complete lack of understanding. More on this topic in the next section; let us for now accept this fundamental limitation of automatic tutors.
Suitable topics for CAI presentation tend to fall into two categories. One is characterized by such words as memorization, objective tests. Learning a new vocabulary, testing whether a driver knows the rules of the road, are good examples. The second category consists of topics where understanding seems to be more important than memorization, but requires a lot of practice and experimentation, in addition to thinking. Such topics tend to share several of the following characteristics:

- they are well understood, factual, objective
- they are rather basic, not in a state of rapid development
- often they can be formulated in a mathematical way
- they exhibit a great deal of regularity
- they are governed by a small number of fundamental laws

The properties above make a body of knowledge "algorithmically tractable", which is the computer scientist's way of saying that the answer to questions can be "computed" - you don't need any intuition. Arithmetic, the grammar of Latin, the structure of molecules, geometric construction, much of elementary physics, or accounting, are perfect examples of algorithmically tractable fields of knowledge. History and geography are borderline cases. If all you want to teach are basic facts, such as "1066: battle of Hastings", then of course a computer will do a good job. But if we teach history at all, it must be because we wish the students to understand why there was a battle of Hastings, how it influenced the development of a country. For this purpose it is not at all clear how a computer can help. It can of course present text, in big or small chunks, continuously or interspersed with questions. This teaching strategy gave CAI the bad reputation of electronic page turning in the sixties.

As a rule of thumb, start your CAI project by choosing only topics that fall into one of the two categories described above:

- memorization (drill and practice)
- an algorithmically tractable body of knowledge, where the student can experiment on a simulation model.

4.2 What, to whom, and how?

Anything goes that you can dream up and program.

When you have decided what general topics are suitable for CAI, you are now faced with the specific content of one lesson that you are designing: a unit that the student will normally study in one session. Your first decisions should be completely independent of the medium to be used for delivering this lesson:

- What are the one or two key ideas that I want to get across?
- What concepts, techniques, results do I want to introduce?
- Who is my audience? (the typical student, the most advanced student whose interest I still want to capture, the slowest student who must still be able to profit from this lesson).
- What do I, as an author, want to know about the student's activities as he is interacting with my program?
When you get to the How-questions, "how do I present these ideas to this audience?", the medium slowly begins to enter into consideration. Try the following trick. Pick a medium which has some resemblance to the computer-driven screen, namely your hand- and brain-driven blackboard or sheet of paper. Picture your typical student as vividly as you can, and ask yourself: what would I do if I personally had to explain this topic to this student? You are allowed to talk, scribble and erase on blackboard or paper, and ask questions. And when you ask questions, that's when the computer comes onto the scene.

As you proceed in this imaginary dialog, you must of course keep asking: can I program this? And you will usually find that as long as you talk, gesture, and draw pictures, the programming of this monolog will be straightforward. Often the computer-driven screen will do a much better job at presenting something than the hand-driven chalk or pencil, for example when you present a mathematically defined curve. The bottleneck of your imaginary dialog shows up when you ask questions, when the monolog turns into a dialog. As you try to anticipate the student's possible responses to your question, the limitations of what you can program show up starkly and painfully. You will find that you must abandon your favorite questions: "Now why do you think this happened?", "What would go wrong if we did it that way?", "Can you formulate a general rule that covers these examples?", "There is an error in this argument; can you tell me where it is?" - because there is hardly ever a chance that your program could give an adequate response to the student's answer. You can of course pose rhetorical questions of this general type, which are then answered by your program after ignoring the student's answer, but that is not a good practice (see section 4.4 "The tutor should not pretend to be smarter than it is").

"Judging the student's response" is the crux of what you can and cannot do in CAI. You can easily judge true/false or multiple-choice questions, a fact which, by logic that puts the cart before the horse, gave a boost to the programmed instruction movement: if you can do it, it must be right. You can easily judge numerical answers, taking into account any tolerance for errors that seems appropriate to you. It is still straightforward, although laborious, to judge numerical answers with units, e.g. 1 Australian dollar = 2 Swiss francs = 1.2 $US. It is straightforward to judge single words or short phrases as they might occur in a vocabulary drill; it is questionable whether it makes sense to try to judge entire sentences. Your program may ask the student to translate a sentence into French. When the sentence is short, there may be only a few straightforward translations of it; but an average sentence from a text has more translations that are acceptable than a program can judge (try to translate this one!).

Your ability to program adequate judging of student responses determines the "How?" of your lesson design. It also clarifies the discussion in the preceding section on "Choice of topic". Bodies of knowledge that are algorithmically tractable are exactly those where apparently complex answers can be judged accurately, adequately. "Draw the curve \( y = \frac{\sin x}{x} \) on this grid", "How
many operations will this program perform as a function of \( N \)? "Fill out this balance sheet", "Draw the shape of the deflected beam when this force is applied" - these are all tasks that require understanding on the student’s part, not memorization, and where a program can provide a helpful diagnosis to most student responses, including wrong ones.

4.3 Setting the level of ambition: smart or dumb tutor?

As a beginning author gains experience and confidence in his skill, he is tempted to write smarter and smarter teaching programs, often called tutors. As we have seen in the previous section, this typically means attempting to judge a greater variety of student responses, in particular, "free-form" or "constructed" responses, as the PI jargon calls them. But the word "free-form" does not catch the essence of what is going on when an author starts expanding the variety of student responses to which he wishes his program to give sensible judgements. Researchers in artificial intelligence have a better word for it. They call it: expanding the program’s domain of discourse.

One must distinguish two different ways in which a program can allow a variety of student responses to be recognized. The easy way is to simply introduce synonyms: ‘yes’, ‘yes’, ‘YES’, ‘sure’, ‘of course’, ‘oui’. Sometimes this is convenient, sometimes it is a pretense of eloquence which is not backed up by insight (see next section). But in any case it is not a fundamental enrichment of the tutor’s capabilities; even when the synonyms involve a little mathematics, such as \( 1 \text{ foot} = 12 \text{ inches} \).

The hard way to enlarge the range of student responses is to let the student express new things, not just express the same thing in different ways. Although it may be hard to define this difference exactly, our intuitive feeling is accurate enough to distinguish whether or not something new is being said. And it makes a lot of difference for the size of the program. If you wish to allow Roman numerals in addition to decimal notation as input, you let the student say the same things in two different ways. One conversion routine will accomplish this extension. If you are discussing the historical development of number notations, and you decide that Roman numerals are also to be discussed, then you truly extend the domain of discourse, and no conversion routine will do the job.

This lengthy preamble is to warn the author that, when he decides to incorporate a small model of reality into his program, to explicitly represent some small body of knowledge that allows him to judge a wide variety of student responses, it had better be really small. The effort, size of data collection, and size of program needed grows extremely fast with the size of the domain of discourse. Research in artificial intelligence shows that programs that can carry out a passable conversation about the geography of South America, or the top-down development of a program to symbolically differentiate an arithmetic expression, easily require tens of thousands of lines.
If your goal is not research in artificial intelligence, but rather development of courseware, don’t attempt to write a smart tutor. You may find that a dumb tutor is a smarter move.

4.4 The tutor should not pretend to be smarter than it is

Beginning authors of instructional dialogs must re-learn the lesson that someone who pretends to possess more knowledge than he actually has will soon be mistrusted. This holds for people as well as for programs. An automatic tutor should make it clear what domain of discourse it can handle. The author must be aware of the danger that careless wording may give the student an exaggerated expectation of the program’s abilities, followed by disappointment when the expectation is not met.

This danger can take subtle forms. When a program acknowledges a correct student answer by randomly choosing among such phrases as "great!", "terrific", "now you really got it", the student may be led to believe that these phrases represent different levels of approval, and interpret a mere "good" as half a failure. But in all likelihood this program can only distinguish right from wrong answers. The author’s mistaken desire to avoid repetition leads to the program’s implicit pretense that it can differentiate more finely than a simple yes or no. It is far better to have this program acknowledge all answers with the repetitious but honest words "ok" or "no".

4.5 Coercion vs. laissez-faire, or:
the student is smarter than your program

One of the legacies of programmed instruction that must be overcome is a tendency towards excessive control over the student’s movement through the material being presented to him. Remember that the word "programmed" in PI has nothing to do with computer programming. It means that the author of the material has programmed the steps or actions that the student must go through to achieve the desired state. It is the student that is being programmed and treated like a robot. For some learning activities this is an efficient teaching strategy; for example, learning certain muscular skills, or memorization. For others it is inappropriate: e.g. understanding a mathematical proof, as opposed to memorizing it.

Above all, PI and all teaching strategies that rigidly enforce a predetermined sequence of actions are ineffective when the student resents this control. Whether his resentment is caused by emotion (he considers this control to be an insult to his intelligence) or logic (for example, he may only need one specific item of information out of the whole lesson, but the program forces him to solve ten trivial exercises first) is irrelevant - you’ve lost the student once he starts cussing at the damn machine.
Except in a few cases (perhaps tests), the library, the book, and the laboratory are much better paradigms for designing CAI material than PI. A good library lets you walk around the stacks and provides a map (catalog) so you can do this intelligently. A book lets you skip chapters or back up, and provides a table of contents and an index so you can do this intelligently. A laboratory provides equipment and manuals for you to make meaningful experiments. A CAI lesson should similarly exploit the user's intelligence, rather than insult it, by letting him make the final choice of steps and actions to be done at any moment. And it must provide the information for the user to make this choice intelligently.

4.6 Tone of conversation:

   a picture (and other things) is worth a thousand words

Text is verbose because the printed page lacks dynamic forms of expression (although cartoons try to imitate animation). A computer-driven graphics screen can use the elements of time and animation to express many things compactly, unambiguously, in a form more rapidly understood by the reader, than a paragraph of text could. Don't say "the sinusoidal curve in the upper right corner of figure 13". Flash that curve instead. Moreover, figures don't need to be numbered. When you want to refer to it, just reproduce that figure.

There is no need to invent long-winded descriptions of processes in action. "A point on the circumference of a rolling wheel moves slowly when it touches the ground, fast when it is farthest from the ground". What a clumsy and inaccurate description. Let a wheel roll across the screen instead, and trace the cycloid generated by a fixed point on its circumference in "real time".

"A picture is worth a thousand words" is a saying invented to describe the possibilities of a static page. To capture the expressive power of a dynamic screen, it must be extended: a picture, proper use of timing, animation allow you to keep the screen (and sometimes the user's mind) uncluttered with verbage.
The good craftsman is known by his tools.

The most brilliant strategic design comes to naught if the reader stumbles over "minor" difficulties at every step. Once the "computer-driven screen" has become familiar to most programmers as a medium for communication, it will not be necessary to point out the fairly obvious rules of good style collected here. In today's man-machine dialogs, however, violations of these rules abound: poor layout of the screen, crowded screen, dead-ends, letting the user get lost in a maze, and many others.

The following brief collection of Dos and Don'ts is intended to make the beginning author of instructional dialogs aware that there are elementary rules of style that must be observed when using the computer-driven screen, as there are for any other medium. The printed page, with its lack of ability for interaction, is a poor medium for demonstrating these rules. Therefore they are formulated briefly, and perhaps not always convincingly. In my PLATO lesson "style" [Ni 77] a more extensive set of Dos and Don'ts is illustrated by letting the user experience the consequences of an author adhering to or violating these rules of style; the user gets the message quickly. In reading this chapter, try to imagine that you are sitting in front of a terminal, where what is merely being described is actually happening to you.

5.1 Know your medium: the computer-driven screen

When using any new medium you must ask yourself: what can it do well, and what does it do poorly? For example, an overhead projector is well suited for superposing several pictures in different combinations; on a sheet of paper this is practically impossible.

The computer-driven screen is good for rapid and accurate presentation of information that can be deduced by means of straightforward algorithms from large amounts of data by means of lengthy computations. It can do so in response to a wide variety of user inputs, as long as this variety is contained in an algorithmically tractable, narrow domain of discourse. It is not adept at tasks that, in human terms, require judgement, experience, insight. By comparison, a teacher at a blackboard is neither accurate nor rapid, nor can he call upon large amounts of data or lengthy computations. We hope he has good judgement, experience, and insight. Books and films may present accurately and rapidly results based on much data and on lengthy computations, but they lack the ability to react meaningfully to a user's input.
In addition to knowing the fundamental abilities and limitations of your medium, you must also know specific details. How many lines and how many characters per line fit on the screen? Use all of them. The student often needs to see a certain collection of items together, at a glance, or it will be difficult for him to "see the whole picture". Can the whole screen be changed instantaneously? If so, you may attempt fast animations; if not, beware of boring the user with slow, long-lasting displays that carry little information. Do you have a color terminal? If so, use color to help the user distinguish the various types of information that coexist on the screen; consistently write messages that guide the user through the lesson ("Press DATA to review the problem statement") in the same unobtrusive color, in the same area of the screen; write background information in one color, questions or requests for student input in another.

5.2 What we can learn from other media

"Enter a date >"
'Jan 1, 1984'
"Out of range"

Many a computer dialog greets the user with a request for which he is totally unprepared. Every book starts with a title page that serves as the first introduction to the book; there usually follows a preface, from which the reader gets some more information before he decides to plunge into the body of the text. There is a table of contents and an index to help the reader find his way around the book, so he can browse as well as read from beginning to end. These elementary facilities must apparently be rediscovered by most authors of instructional dialogs.

The graphic arts teach us that the human mind understands the information displayed on a page fastest if the layout is balanced, the page is not crowded, and items that are related are placed near to each other. The computer community apparently needs to learn the simple rule that esthetics is related to understandability. The typical computer dialog uses a video display as if it was a teletype, appending the current input or output to a solid block of previous discourse, much of which is no longer of interest to the user (in particular, erroneous input), and thus clutters the screen and distracts attention.

An amateur camera-man's film gives us a headache when he sweeps a landscape too fast for the eye to follow. Beginning authors of computer dialogs tend to program animations so fast that the bewildered user only sees a flash in a corner of the screen and wonders what information he has missed.

An author of instructional dialogs does well to read books, watch TV, and listen to radio or podium speakers with deliberate attention to the techniques they use to give their audience an overview of the entire presentation, and to highlight points of particular importance. George Polya, a famous mathematician and teacher, used to say "the teacher is an actor". So is the author
of instructional material, indirectly through his medium.

5.3 Educational hang-ups

1) On control and reinforcing

I have already mentioned repeatedly that authors of instructional dialogs have to overcome a tradition of excessively controlling the student's path through a lesson. Exercising control can take much more subtle forms than giving the student no choice. A hidden form frequently observed is to give the user a choice, but without the information necessary to make an intelligent choice. The question "Would you like an opportunity to practice?" appears to give the student a lot of freedom, but in fact he does not know at this point what will happen to him if he says yes, no, or maybe. He can find out only by time-consuming trial and error, and thus is not really in control of the dialog. The instructions "Press E for a set of exercises, NEXT to proceed to the next topic" put the user in control. "Say it, don't ask" may be a way to remember that a question does not carry as much information as an instruction.

According to PI theory, the student should be rewarded (reinforced) for correct answers. Since CAI systems usually can't dispense candies, many authors attempt to reward the student verbally: "Here are some stars for you ** *". The shallowness of this kind of reward turns many people off. Omit it. Hold the dialog in a concise and factual tone.

2) Don't collect more data than you are willing to look at.

The computer allows you to collect cheaply data about almost any measurable aspect of the student's behaviour: how much time he was logged in (but not how much time he was paying attention), how many answers he got right or wrong (but not whether he actually tried to solve the problem or was merely guessing), how long it took him to answer (but not how long he had been thinking). No other medium allows the author to get feedback so quickly and comprehensively. The author should use this possibility of obtaining feedback, and validate his lesson by observing how it fares when exposed to its typical audience.

Data is of no use unless you look at it and act upon it. The hard question "what data will cause me to revise my lesson?" should be answered before the data has been collected, otherwise the temptation is great to collect so much data that the little that could make a difference gets buried in the mass of irrelevancy.

In general, data about elapsed time is of little interest, except in a drill where speed is an explicit goal. Our thoughts often wander off the immediate task at hand, for better or worse pursuits, and what the author meant to measure (time devoted to the task he posed) is not what the clock tells him.
Data that tells the author whether he has correctly anticipated the user’s state of mind is useful to collect. For example, if the author expects a free-form answer, he should definitely collect actual student answers during the early life of his lesson; he may be surprised at the number of answers that he had not anticipated. Similarly, if the author attempts to diagnose the student’s thought process through the answer, he should collect actual student answers, and if possible interview some students to verify that the relationship between reasoning and answer is as he expects it to be.

Above all, the author should check whether his instructions and formulations are clear. Giving the student a convenient opportunity to make comments right at the moment when something is puzzling him, without having to exit from the lesson, is the most important contribution to meaningful data collection that a CAI system can provide.

5.4 Programmers’ hang-ups

1) Clock control is fun, but user control is better.

A programmer who is given the opportunity to program interactive graphics dialogs is likely to become enamored with this new toy, and to explore its full range of expressive capability. That’s fine as long as he keeps his toy programs separate from the instructional dialogs he writes. Painting a picture on the screen and removing it under clock control may create a dazzling show, but is likely to infuriate the student who wants to look at it at his own pace.

2) I know you can parse it, but there should be no need to.

"How many apples and oranges can you buy for a buck?" The programmer may be proud of the fact that his program can make sense of such varied answers as "two apples and 13.0 oranges", "one each", "2 RETURN 5 RETURN". But the question is poorly phrased, and the student will waste time guessing how he should formulate his answer. "Enter the number of apples >" followed by "Enter the number of oranges >" is a much clearer description of the requested student answer.

3) Defaults and options.

In computer jargon these two terms denote an attempt by a designer to catch in one basket a wide range of user choices, each of these choices being specified by a list of parameters. To save the user typing, default values are introduced for parameters whose values have not explicitly been specified.

This style of dialog is efficient for a trained user who knows what he wants when he approaches the computer. It is inappropriate for the casual user, who is bewildered by the multitude of options presented to him, and feels insecure about whether a short answer (with default values) or a long one (all parameters specified) is expected of him. The student at a CAI system is
usually a casual user, who never practices long enough with any one component of the system to become an expert user. Any command language designed for such users must be simple, normally offering a choice among only a few alternatives at each step.

As a designer of instructional dialogs, forget your options and defaults, and concentrate on identifying the minimal set of commands that the user needs.

5.5 The dynamic page: don’t scroll or scramble

One of the mind’s powerful techniques for organizing a wealth of material, well-known as an aid to memorization, is spatial analogy (see, for example [Be 73]). When we wish to imprint on memory a train of thoughts, we associate with each of these thoughts a place in a familiar environment. In order to recollect the sequence of ideas, we imagine walking through this environment, picking up the thoughts in the places where we had deposited them earlier. We use spatial organization when we accumulate incoming mail on one corner of the desk, outgoing mail on another, and pile up reports in a third spot.

An effective dialog uses the screen so as to make it easy for the user to sort out the different types of information that are usually present at the same time: logistics information, such as an indication of where in the lesson the user is, what he is supposed to be doing there, how to get help if he needs it; background information or reminders that he may need to solve his problem ("force = mass \times acceleration"); the student’s last input, if it is still relevant to the current topic. All of these are consistently written in the same area of the screen, so the user knows immediately where to look for any kind of information he may want.

The teletype and other printing terminals do not conveniently allow space on the roll of paper to carry any meaning other than the time-sequence in which input/output transactions occurred. This restriction has led to a dialog style called "scrolling", which is appropriate for teletypes but does not exploit the possibilities of the video terminal.

A dialog to be delivered by a video terminal should be organized in terms of dynamic pages. There is a well-defined moment when a page appears on the screen, and when it disappears, i.e. is replaced by another page. Each page has a name, perhaps of mnemonic value (e.g. "control panel"), perhaps just a unique identifier, provided it carries some meaning, such as "exercise 3"; this name is displayed in a corner of the screen for the entire duration of the page. Some other information also remains unchanged, such as a brief statement of the purpose of this page. Much of the page may vary dynamically. For example, if the user turns simulated knobs on a simulated control panel, then a portion of the screen must be reserved to continuously display the current state of the simulated system.
A dynamic page serves as a useful organizational unit of courseware if it is devoted to one or a few related concepts, and if the user's activities are similar during the entire duration of the page. A page should never suddenly disappear, "slip away from under the user". A conscious action on the user's part is needed to turn a page, and to insure this, the system should give explicit notice of termination (such as "Press NEXT to leave").

As an organizational unit, a page can be compared to a paragraph or section in a book, or to a traditional "frame" in programmed instruction. The difference between a dynamic page and a PI frame is that we have in no way restricted the content of a page. The content is what makes an instructional dialog; structuring this dialog into pages is just a convenient way to give the user control over the dialog.

5.6 Structure, maps, and (fast) motion

The dynamic page is a structuring technique appropriate to small units of material. A lesson intended to be covered by the student in one session easily contains a dozen pages: a title page, a table of contents, a few tutorial pages where background information is being presented, a few pages where the student can manipulate a simulation model, some exercise pages, and a summary page. A course contains hundreds of pages. In order for the student to understand the organization of the course, and to be able to study it efficiently, the course designer must impose a visible structure over this large set of pages.

Observation of hundreds of students at CAI systems (and of casual users of interactive systems designed for other purposes), has shown to me that the difficulties most frequently experienced are well characterized by the following questions:
Where am I? What can I do here? How did I get here? (in case the user is surprised at the system's behavior) Where else can I go, and how do I get there?

A well-structured dialog must provide conveniently available answers to these frequent user questions. The concept of dynamic page, and the name given to each page, provide an answer to the question "where am I?". "What can I do here?" should be answered by displaying a list of all commands that are active at the moment of inquiry, i.e. are active on the current page. This list should not be long: the entire set of commands is best partitioned into "modes", i.e. subsets that allow a user to perform an identifiable task, such as editing. On a page where the user is expected to draw a picture, only the commands of the mode "picture editing" need be active. Commands of the mode "execute programs" are useless on this page, and if active, would only contribute a potential source of errors. As the user's activity is restricted on a page, so should be the set of active commands.

There is one particular mode, or set of commands, that must be active at all times: the motion commands. Motion commands allow the user to leave a page and move to any other page in the lesson or in the course. This motion must be fast or slow as the user
desired. If he wishes to inspect page after page in a leisurely browsing mode, a motion command NEXT PAGE suffices. If he is seeking some specific item of information, then he wants fast access to the 99-th page, and this requires additional motion commands. In case he knows what page he wants, a direct-access command GO TO page so-and-so suffices. Often he does not know the name of the page that contains the desired information, but only some vague description such as "an exercise in the chapter on matrices". For this case we need fast motion commands that skip from chapter to chapter, without forcing the user to look at all the intermediate pages.

There are many ways to implement a consistent and complete set of motion commands in a lesson, course, or entire interactive system. The set of commands will depend on the structure imposed on the set of pages. In a hierarchical structure a few commands that move from any node in the tree to its predecessor, left and right sibling, and any of its direct descendants are sufficient to allow the user to move from any page to any other in at most half a dozen single-key-press commands. Whatever structure is imposed on the space of pages, the user must be able to see a map of his current neighborhood at any time.

In a well-structured space of pages with a complete set of motion commands, the question "where else can I go, and how do I get there?" can be answered once and for all: the same motion commands apply throughout the system, and always have the same effect (see Nievergelt and Weydert [NW 79] for a more technical discussion of these ideas).

There remains the question "how did I get here?" when the user has mistyped or misunderstood a command. In this case, he often wants to return to the page he came from, to undo the last step, including any actions that may have changed data. A system should store a part of the user's "trail" as he is interacting with it, at least as much as is necessary to return him to the previous page, including the state of all data that he had at that point.
Exaggerated promises and expectations have been commonplace in automated education since the early sixties, when the first generation of CAI systems gave the dormant field of teaching machines a new direction. Hence it is tempting to dismiss yet another prediction of a breakthrough as optimism unsupported by facts. Indeed, experience with CAI is sufficiently recent and inconclusive that the history of this technology can be interpreted equally well as a promising start on the problem of finding the right way to exploit a powerful new medium, or, at the other extreme, as a sequence of failures, each failure followed by a redefinition of the problem by those unwilling to draw the inevitable conclusion.

I have already mentioned that I consider the fundamental novelty that CAI introduces to have little to do with education. The essence of this novel technology is that we have a powerful new medium, with properties radically different from other media. It is the only two-way mass communication medium we know. We don't ordinarily talk back to the newspaper or the TV set, and only a few listeners in a large audience get to talk back to the speaker; but we do expect that every student who follows a CAI course talks back to the program a lot—at least within the limited domain of discourse it is able to handle. Education is involved only to the extent that the educator now has a new option in his choice of media.

It is instructive to look at the traditional media, and to compare how long it took to make the transition from a prototype that demonstrates technical feasibility to an established commercial product. Here are some relevant dates.

<table>
<thead>
<tr>
<th>Medium</th>
<th>Date</th>
<th>Event or Invention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>1455 Gutenberg's bible; first printed book</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>printing shops in every major city of Europe</td>
</tr>
<tr>
<td>Telegraphy</td>
<td>1836</td>
<td>Morse's invention</td>
</tr>
<tr>
<td></td>
<td>1866</td>
<td>first transatlantic cable</td>
</tr>
<tr>
<td>Black/White</td>
<td>1923</td>
<td>Zworykin patent (iconoscope)</td>
</tr>
<tr>
<td>Television</td>
<td>1950</td>
<td>widespread use</td>
</tr>
<tr>
<td>Color Television</td>
<td>1940</td>
<td>Goldmark's rotating filters</td>
</tr>
<tr>
<td></td>
<td>1945</td>
<td>fully electronic</td>
</tr>
<tr>
<td></td>
<td>1962</td>
<td>RCA's first year of profit on color TV</td>
</tr>
</tbody>
</table>
The transition period appears to have varied from 25 to 50 years. If we take the mid-sixties as a representative date by which the technical feasibility of the computer as a medium for delivering instructional dialogs had been demonstrated, then, according to the analogy above, we must not be surprised that commercial success has eluded CAI so far. Two decades, which appear to be so long in a rapidly moving field such as computers, is just a growing-up period in comparison with the development time required by other media.

While this analogy with other media is not compelling, it does strengthen the resolve of the optimists among us, and gives some reason to bet that CAI will be an established addendum to our educational repertoire within our lifetime, or perhaps within a decade.
The history of mechanical teaching machines, to the extent that it influenced the thinking of the pioneers of CAI, can be traced to Sidney Pressey. He built machines for automatic testing and scoring, and observed that this is one form of teaching, as the title of one of his papers indicates:

[Pr 26] Pressey, S. L.
"A simple apparatus which gives tests and scores and teaches" School and Society, Vol 23, 373-376, 1926

Two classical papers from the period of development of programmed instruction as a science, are:

[Sk 54] Skinner, B. F.

[Cr 59] Crowder, N. A.

The history of this early phase is well presented in:

[Fr 63] Fry, E. B.

During the sixties a number of CAI projects sprang up, often based on the idea that a computer is the ideal delivery vehicle for drill and programmed instruction. Representative papers from this decade are collected in:


An influential CAI project, with heavy emphasis on educational research, is documented in:

[SM 72] Suppes, P. and Morningstar, M.

Another early CAI project, which has undergone continuous development since 1960 and is today the largest such project in existence, is PLATO. To assess the development over a span of almost two decades, compare the following two papers:

[BB 61] Bitzer, D. L, and Braunfeld, P. and Lichtenberger, W.
[Bi 77] Bitzer, D. L.

The production of courseware on an industrial basis in a large scale CAI project, TICCIT, is described in

[Bu 72] Bunderson, C. V.

The reaction to electronic page turning is amusingly expressed in

[Lu 72] Luehrman, A. W.
"Should the computer teach the student, or vice-versa?" AFIPS Conf. Proc., Vol 40, 407-410, 1972

The LOGO project, one of whose main goals is to turn the computer into a powerful tool for children to control when exploring the world of mathematics, has influenced many people in the fields of computers and education. A typical task is to program a computer-driven "turtle" to drive around the floor or draw interesting figures. A representative account is given in

[Pa 70] Papert, S.

and also in
Programmed Learning and Educational Technology, Sep 1972, 245-255

A project with similar goals but much more powerful hardware is the Smalltalk system at Xerox PARC

[Ka 72] Kay, A.

An account of the synthesis between "computer teaching student" and "student teaching computer", expressed through the analogy of "dual mode" and "solo mode" in airplane flying, can be found in

[Dw 71] Dwyer, T. A.
An earlier account of the development of CAI, and a case study of a comprehensive CAI laboratory for teaching computer science, is

[NI 75] Nievergelt, J.  

Another recent survey of CAI, including a history and an assessment of the current state and trends, is

[He 77] Hebenstreit, J.  

An overview of the variety of typical CAI applications in active use today can be obtained from the following papers:

[Bo 78] Bork, A.  
"Machines for computer-assisted learning" Educational Technology, Vol 18, No 4, Apr 1978

[Fe 75] Fletcher, J. D.  

[SS 76] Smith, S. G. and Sherwood, B. A.  
"Educational uses of the PLATO computer system" Science, Vol 192, 344-352, 23 Apr 1976

[Si 78] Zinn, K. L.  
"Sources of information about computing in instruction" Educational Technology, Vol 18, No 4, Apr 1978

[IB ] IBM  
"Field instruction system"

[SC 74] Scanlon, R. G. and Connolly, J. A.  

Representative research projects in the application of artificial intelligence techniques to CAI are summarized in

International Journal of Man-Machine Studies, Vol 11, No 1, (Special Issue on Intelligent Tutoring Systems), Jan 1979
A comprehensive study of costs and benefits of computer uses in education can be found in

[Le 72] Levien, R. E.
"The emerging technology: instructional uses of the computer in higher education", McGraw-Hill, 1972, 585p

The existing literature on the design of man-machine dialogs is scarce. A good collection of case studies can be found in the book

[Ma ??] Martin, J.

Readers who have access to the PLATO system may be interested in the lesson

[Ni 77] Nievergelt, J.
"Style: some thoughts on style and techniques of lesson writing" Control Data Education Company, Minneapolis, 1977

Systematic approaches to the design of man-machine dialogs, based on spatial models of the set of data and the set of commands available to the user, are outlined in

[Be 78] Bennett, J. L.

[NW 79] Nievergelt, J. and Weydert, J.
"Sites, modes, and trails: telling the user of an interactive system where he is, what he can do, and how to get places" Proc. IFIP Conf. on Methodology of Interaction, Seillac 1979, North Holland Publishing Co.

The existing literature on CAI is primarily a collection of case studies, of the type "this is how we did it". There are hardly any "classical papers" that introduced a drastically new idea or approach in this field - most projects come to similar conclusions. It is useful for the newcomer to CAI to read a dozen of these papers, but it does not matter much which ones. The papers referenced above appear to me to be typical of the literature of CAI as a whole. An extensive bibliography of the CAI literature up to 1973 is

[TJ 73] Testerman, J. D. and Jackson, J.
Acknowledgment
-------------

I am grateful to the following people for helpful comments which have improved this paper:

The University of Wollongong provided the opportunity to write this survey and introduction to CAI by inviting me to lecture on this topic.

In parts of this paper I have drawn on two earlier publications of mine. Section 2 on the history of CAI is adapted and updated from [Ni 75]. Section 5 contains some material which has been more vividly presented in my PLATO lesson "Style: some thoughts on style and techniques of lesson writing", Control Data Education Company, 1977.