THE JOURNAL OF THE LEARNING SCIENCES, 1(1), 7-35

Empowering the Student: New Perspectives on the Design of Teaching Systems

Roger C. Schank
and
Menachem Y. Jona
Institute for the Learning Sciences
Northwestern University

Most efforts in making computer-based teachers have not tried to change the standard teaching methods commonly used in schools today to any significant degree. We survey several of these methods of teaching and examine both their strengths and weaknesses. We then introduce six new *teaching architectures,* defined as a kind of general blueprint or framework that embodies a particular approach to teaching. We argue that using the power of the computer-based teaching medium, such teaching architectures can be designed to exploit the strengths of standard teaching methods while avoiding many of the pitfalls.

Finally, we present a perspective on the role of computers in education, arguing that one of the main problems facing education today is students lack of control over the instruction they receive. As a first step in remedying this problem, we propose that educational software should provide students with a set of "teaching buttons" that will allow them to communicate easily with a computer-based teacher and thus more fully gain control over their interaction with the system. An example of the current version of our set of teaching buttons is also presented.

The world has changed, this much seems obvious. On the other hand, school has not changed. Despite major differences between the world of the early 1990s and our world today, school is frighteningly the same. This

Requests for reprints should be sent to Roger C. Schank, The Institute for the Learning Sciences, 1890 Maple Avenue, 3rd Floor, Evanston, IL 60201.

might be an acceptable state of affairs if school had been so well thought out in 1900 that no important changes were necessary, but this is hardly the case. The mission of the schools of the 1900s was primarily to train good factory workers, not well educated citizens (Apple, 1979).

Many current teaching practices arose from technological factors and practical constraints that no longer apply (Schank & Edelson, 1990). The technology of the book has long directed what is meant by education. Furthermore, a scarcity of books, both in diversity and in number, contributed to the installation of the lecture method of teaching and the concept that school work outside the classroom meant more books. Computers can change the educational scene by allowing us to reconsider what it means to learn and what it means to teach. The power of the computer has provided us with a blank canvas on which to draw a new picture of education. Freed from the practical constraints that have limited teaching in the past, we can now, to a much greater extent, tailor teaching to the needs of the individual student and to the particular subject matter being taught.

But how can teaching be changed? In answering this question, it is worthwhile to characterize teaching practices currently in use so that we may see where the problems are and, hence, where there is room for improvement. In this article, our goal is twofold. First, we attempt to characterize the teaching practices currently in use. Second, we present some new methods of teaching, new teaching architectures, that capitalize on the capabilities of the computer to overcome many of the limitations of conventional teaching methods.

METHODS OF TEACHING

We begin by looking at six teaching methods that are commonly employed by teachers today. These are meant to represent a survey of some of the more widely used teaching methods, rather than an exhaustive catalog.

The Sponge Method

The standard method of teaching is what can be termed the *sponge method.* This method views the student as a kind of sponge (i.e., a passive absorber of information). This method is based on the notion that a student is capable of learning simply by being presented with information. Expecting students to learn things by being told about them has been a widely accepted idea, even making its way into early education research. For example, Carroll (1968) began by stating: "By far the largest amount of teaching activity in educational settings involves telling things to students . . ." (p. 4).

The sponge method is not rooted in any theory of learning, but rather in practical necessity. In most educational settings students far outnumber the teachers, creating situations where a third-grade teacher's classroom contains 30 pupils, or a college professor must lecture to an auditorium filled with 500 freshman. With this kind of mass education, all that can be done is to present information and perhaps answer a question or two. Interaction of a meaningful kind is difficult, if not impossible, to achieve.

There are several problems with the sponge method. It is most often used when the audience is expected to be passive. This is an unfortunately common situation, seen from *Sesame Street* to lecture halls in universities. Notwithstanding the passivity of the learners, this method is not completely useless. People can and do learn this way. The problem is that what is being learned by each person is much more idiosyncratic than might be thought. People are unable to absorb everything that is presented to them while they are passively "absorbing"—their own thoughts intervene. They attempt to relate what they are hearing to what they already know (see Bransford, 1979; Schank, 1982; Schank & Abelson, 1977; for a fuller discussion of how this works). Because each listener's knowledge is different, each absorbs something different. This nonuniformity is not necessarily a bad thing, in fact it is quite valuable, unfortunately the sponge method of teaching fails to exploit it.

A second problem with the sponge method is that even when it succeeds in having students absorb information, it encourages students to take information as gospel, without questioning. Ideas cannot be easily challenged in this format of education, as any student who has tried to disagree with his professor during a 500-person lecture will quickly attest. Even when the teacher does provide alternative points of view, the student will often simply absorb the views associated with each perspective without bothering to evaluate, compare, and contrast the views himself. This is related to the problem of "inert knowledge" (Bereiter & Scardamalia, 1985; Bransford, Franks, Vye, & Sherwood, 1989; A. L. Brown & Campione, 1981).

A third problem with this common teaching method is the role it assigns to the learner. The sponge method implicitly assumes that the world contains a vast store of information, and that the learner's task is to absorb as much as possible. Once this view is internalized, teaching becomes defined as simply knowledge transmission. The teacher's role is one of a teller of truths. The student is expected to absorb, or memorize, these truths and to be able to recite them when necessary.

If the sponge method is so problematic, how is it that people often seem able to learn via this method? Despite its many problems, it turns out that sometimes the sponge method is exactly the right method of teaching. When is this? When students want to know something or when they have a question, they are ready to hear what it is we are about to tell them. When students have become curious and want an answer, telling them the answer works fine much of the time. Thus, the trick to making the sponge method work is to make sure that students are ready and eager to hear what the teacher is about to tell them.

The Apprenticeship Method

Another popular method of teaching is the *apprenticeship method.* This method borrows from the one used throughout history to teach vocational skills from hunting to watchmaking to car repair. In this method, the student watches the teacher perform some task and then tries to master the required skills under the watchful eye of the teacher. This method has been quite successful in teaching real-world, on-the-job skills; unfortunately, in the transfer to the classroom, the fundamentals of this method have been forgotten, and the net result has been far less successful.

There are several serious problems with the apprenticeship method, the first is that it can easily degenerate into a rote learning task. Often the way the apprenticeship method is implemented is that the student is told exactly how to perform some task, given the rules to follow, shown how to do each step, and rewarded when the correct set of steps is performed. Taught this way, the student is being encouraged to conform, rather than to understand the task. For example, Schoenfeld (1985) found evidence that problem-solving skills learned in a rote fashion are tied to the surface features of the problems rather than being fully understood. Instead of relying on knowledge of problem-solving strategies or intrinsic properties of the problem, students were found to rely on knowledge of standard patterns of problem presentation common in textbooks. To avoid such rote-learning effects, it is essential that help be provided to students only when they are at an impasse, and then only as little as necessary to get them to complete the current task (J. S. Brown & VanLehn, 1980).

A second problem with this method of teaching is when the real skills to be taught, such as reading and arithmetic, are subdivided into a set of smaller, and presumably easier, skills. Although this divide-and-conquer technique may seem like an effective approach, there are two ways in which it may not work well. Some tasks do not lend themselves easily to being broken up into pieces, that is, we may not know what subskills comprise the larger skills. Second, the hidden burden of the divide-and-conquer approach is that the individual pieces must be put back together again. Often in concentrating on learning each subskill, the student may not learn how each fits into the larger task, and the motivation for learning the subskills in the first place becomes clouded. In other words, the forest gets lost for the trees. J. S. Brown (1985) provided an example of an attempt to rectify this type of problem. Students learning algebra who lacked skills in carrying out low-level computations were relieved of the burden of that task in favor of letting them concentrate on higher order reasoning and strategies needed to solve interesting problems.

The third serious problem with the apprenticeship method is best understood by an illustration of when the method works well. The apprenticeship method is used with some success in graduate schools. There, students "sit at the feet of the master" watching what their advisor does and trying to do it too. The reason the method works so well in this situation is that graduate students are trying to learn to be researchers or professors by watching their advisor be a researcher or professor. In other words, the apprentices are watching the master do exactly the job they are trying to learn. Third graders, on the other hand, are not apprenticing to be third-grade teachers. The apprenticeship method is most successful when working with someone who does what you are trying to learn; it is least successful when a teacher is put in the position of teaching skills they do not know well.

Cognitive apprenticeship theory (Collins, J. S. Brown, & Newman, 1989) is one attempt at reformulating the apprenticeship method that avoids many of the problems just mentioned. According to Collins et al., the techniques of modeling, coaching, scaffolding, and fading are borrowed from traditional apprenticeship practices and applied to the teaching of cognitive skills like reading, writing, and mathematics.

The Artist Method

A much less frequently used method of teaching is the *artist method.* In this method the goal is to stimulate creativity as much as possible. To this end, the way this method is typically employed is by giving students a "blank canvas" on which to paint. Very little guidance is provided because the goal is to teach the student self-expression. The explicit lack of guidance is what sets this method apart from its cousin, the apprenticeship method. Too much guidance about what to "paint" or how to paint indicates that the apprenticeship method is being used.

The artist method has its good attributes. Asking students to be creative is better than asking them to be passive, as in the sponge method. What is missing from the simple description of the aforementioned artist method is the watchful eye of the teacher who is ready to help out if the student gets stuck. The problem is that being a benevolent observer who jumps in with a little bit of advice at just the right moment is extremely difficult to implement for two reasons. First, it requires a tremendous amount of time, especially in classes of 20 or 30 students. A single teacher can not be in 30 places at once, so the teacher may not be available to jump in at exactly the moment the student needs their input. Second, and in many ways more important, is that it is very tempting to want to tell students what to do, rather than helping them discover it on their own, thus turning the artist method into the apprenticeship method. To be able to watch and give occasional hints requires not only a very patient teacher, but one that is able to keep track of what each student is doing and what problems they have encountered.

In order to make the artist method effective, teachers must have the time to devote to monitoring and helping each individual student. The artist method, therefore, lends itself to a one-on-one teaching environment. Until the day that there are enough teachers for each student to have their own, we must look to the promise of such individualized instruction being delivered via computer-based educational systems.

The Research Method

In the *research method* of teaching, students are asked to research some topic or historical event and present a report of their research. This report can take a variety of forms, from the quintessential book report in grade school to full length research papers in high school and college. In many cases, a written report is augmented or replaced by an oral presentation in front of the class. As was the case with the apprenticeship method, this method of teaching, despite its many strong points, meets with far less success than would be expected. Why is this?

The main problem is that these research assignments are often short term, isolated, and irrelevant. When students are encouraged to pursue a subject in depth, especially a subject that is of real interest to them, the exercise can be of great use in teaching. Studies have pointed out the importance of creating situations in which students are motivated by intrinsic interest in a task, as opposed to extrinsic factors like achieving good grades (Lepper & Greene, 1979; Malone, 1981). Encouraging students to find out more about a subject that interests them can truly excite them about learning in general and make them real experts on one subject in particular. Unfortunately, what often happens in actual practice using this method is that students are asked to write a report on a topic that does not interest them. To make matters worse, once they do find something out, the rewarding role of becoming an expert on that topic, so that they can instruct others about it, is often lost in a dull report to be delivered to an uninterested audience.

A second problem, one that is often downplayed or ignored, is that the student can often with very little effort, become more knowledgeable about the subject he or she has chosen to research than the teacher. Many teachers feel uncomfortable being placed in this position. Our society has deeply ingrained the idea that teachers are expected to know more than their students, their authority and effectiveness as a teacher being undermined if they do not. This societal role makes it difficult for many teachers to implement the research method effectively.

From the examination of the problems that are commonly seen with the research method, we can extract the requirements for implementing it successfully. First, it is important for students to have a real interest in the topic being researched, students should be encouraged to pick a topic that interests them rather than having a topic imposed on them. Second, the importance of letting students accrue the rewards of being an expert on the topic should not be overlooked. Students should be allowed to take pride in their expertise and should be encouraged to share that expertise with others (Collins et al., 1989). Third, and perhaps most important, teachers must feel comfortable letting the students surpass their knowledge level (cf. Scardamalia & Bereiter, 1991 – this issue). Teachers who are able to do this, in fact, can use students' inherent pleasure in knowing more than the teacher as an invaluable source of motivation for their students.

The Exploration Method

A teaching method that is currently enjoying widespread use, even outside of the classroom, is the *exploration method*. This method was around before the advent of computerized video games, although that is its most recent incarnation. The notion of the student as an explorer is intuitively appealing: having students figure things out on their own and discovering what they need to know by finding it out for themselves. A version of this method of teaching is used in Montessori schools (Montessori, 1966).

As is the case with the methods discussed previously, the exploration method commonly falls short of the potential of which it is capable. In the computer adventure games, for example, the player is usually called on to guess at what is going on. Although this can be useful in winning the game, it is not exactly the kind of learning that could be achieved. Learning requires a reasonable hypothesis about what is going on in the first place (Schank, 1982, 1986). The exploration method of teaching, as implemented by computer adventure games, means putting the student in a potential learning situation and hoping.

The problem is that hoping is not always enough. As in the artist method, having a teacher around to help out when the student gets stuck, to provide encouragement when the student gets frustrated, or to offer hints about interesting exploration paths when the student gets bored is necessary for this method to reach its full effectiveness. On the other hand, the same problems facing teachers in the artist method (i.e., the need for patience and one-on-one monitoring) are encountered here too. Too much help or too many hints can turn this method into the apprenticeship method.

Thus, effective implementation of the exploration method requires first and foremost that the material to be learned be inherently discoverable. Not all material can be discovered. For the material that is or can be made to be discoverable, the student must be provided with an environment that is truly explorable and interesting that contains that information. In addition, a successful implementation of this method requires a teacher that will "look over the shoulder" of the student, one that can jump in to provide hints and encouragement at the right times, but more important, one that knows when to say nothing so that the student will be able to discover things on his or her own.

The Argument Method

One teaching method that is used in only a few situations, yet still worthy of mention, is the *argument method.* The essence of this method is to force the student to adopt a position on some issue and then defend it. Truth is less valued in this method, in contrast to the sponge method. The goal is to get the student to think on his or her feet, to be critical and analytical, rather than to get at truth. Arguing about things requires thinking about them and trying to outfox an opponent can serve as a real motivation to get students to think deeply about an issue. For example, a study by Smith, Johnson, and Johnson (1981) found that students who studied in a group where discussion and controversy were encouraged showed more accurate understanding, better retention, and higher motivation than students in concurrence-seeking groups or individual study.

The most common uses of the argument method can be found in law, medical, graduate, and professional schools. In fact, the skills of critical analysis, argument, and quick thinking form the basic skill set for students in these schools. The aspect of the argument method that makes it so valuable for these and other situations is that it allows students to test out their ideas. Being able to test out one's ideas is critical for learning. It is no surprise that people so often seek out someone to help them "sound out" their ideas. Testing out ideas on a friend who is playing "devil's advocate" helps elucidate weak spots and omissions which are overlooked when one is working alone (Bloom & Broder, 1950; Frase & Schwartz, 1975; Toulmin, 1958; Whimbey & Lochhead, 1978).

The argument method is a valuable one because, as just mentioned, it teaches critical thinking. The reason that it is so rarely seen, except in the upper echelons of higher education, is that it is extremely difficult to implement in situations with high student–teacher ratios. No teacher can pursue 30 lines of argument at the same time. Students in larger classes will invariably fail to get their argument heard, thus failing to get an opportunity to test out their ideas. Most end up simply being passive observers of the arguments of others. Testing the validity of one's ideas will teach far more than hearing the truths of others.

Thus, to implement the argument method successfully, a teacher must insure that each student is able to test out their ideas, by getting their arguments heard. Furthermore, the teacher can serve a valuable function by modeling good debating and critiquing skills (Collins et al., 1989), perhaps in the role of mediator.

TEACHING ARCHITECTURES

Will computers fix the problems with these common teaching methods? Most efforts in making computer teachers have not tried to change the teaching methods just described in any significant way. Rather, computer programs have tended to institutionalize the aforementioned teaching methods that were inherently computer-like. But, intelligent computers could be intelligent teachers. This will not happen unless we begin to address the issue of what good one-on-one teaching looks like. To do this, we must define new architectures that define the ways that such teaching programs might be structured.

A teaching architecture can be defined as a kind of general blueprint or framework that embodies a particular approach to teaching. Using the power of the computer-based teaching medium, teaching architectures can be designed to exploit the strengths of the standard teaching methods discussed herein while avoiding many of the pitfalls.

Education by computer has almost always entailed a certain explicit teaching architecture, something we can call the "page-turning architecture." This architecture has been left implicit in the design of educational software and is responsible for the lack of excitement in or educational value of most of the educational software on the market today. A page-turning architecture basically involves putting up a screen of information and asking students to either indicate when they want the next page of information or to answer a question that will cause them to get another page of information. The role of the student assumed by this architecture is as a reader of information or a selector of multiple choice answers. Clearly there is a problem here. Such software assumes that learning involves either the passive absorption of information or the recognition of answers to questions.

Certainly we can do better. In the following sections, we present six teaching architectures. These architectures assume that learning involves more than reading and answering. However, if reading and answering are involved, their involvement ought to depend on preparing the student such that reading or answering is precisely what he or she wants to do at the moment that the computer causes that to happen. To do this means treating software somewhat differently. For one thing, it entails enabling computers to store and retrieve far more information than has been typical so far. It also entails representing the information, or at least the content of the information, so that it is accessible at the right time for the right reason. In other words, to make machines better instructors, we must give them better information in an accessible form. Furthermore, we must design teaching architectures that make use of such information.

The six architectures presented here are inspired by what we know of how people learn and the information that we believe can be imparted to machines that will enable them to be better instructors. These architectures are, then, artificial intelligence architectures for the design of teaching systems.

Case-Based Teaching Architecture

The first architecture we discuss, is called the *case-based teaching architecture*. This architecture depends on two ideas. The first is that experts are repositories of cases. When doctors or lawyers are doing their jobs, for example, they are likely to rely on knowledge of previous cases to help make decisions (Bain, 1986; Riesbeck & Schank, 1989). They may recognize a situation as being quite similar to one that has been previously encountered. The task then is to draw on the similarities in the case they have been reminded of, in order to better understand the new case. They would also want to understand the differences to help assess just how relevant the case thought of really is and to help them in thinking about other previously encountered situations that might be more relevant to the current case. This is what we call *case-based reasoning* (Hammond, 1989; Kolodner, 1988; Kolodner & Simpson, 1989; Riesbeck & Schank, 1989). It is clear that much reasoning by experts relies on a significant case base that can be utilized when necessary.

When experts teach, they often rely on a case base. The second key idea is this architecture is that teachers are good storytellers. Or, put another way, in order to relate a case that you know about to students who would learn from it, three things have to be true. First, students have to be ready to hear about the case. Also, they have to be in a position to use the information for something they are working on or interested in. Finally, the case has to be conveyed in such a way that it captures and holds students' interests. In other words, when they are ready and eager to listen to a good story, students are in a position to learn from that story.

The case-based teaching architecture exploits this basic capacity for students to learn from stories and the basic desire of teachers to tell stories that are indicative of their experiences. The task of software that exploited this architecture would be to place students in a situation that they find inherently interesting. Such a situation might involve, for example, having students attempt to build, design, plan, or create something on the computer. This basically creative task should be one that is inherently appealing so that there would no problem arousing interest in the task. The task should be complex enough that all the required information is not immediately available. The task of the program is to teach the student what he or she needs to know, or might consider while doing the task, at precisely the points in the task at which the student becomes interested in knowing this information. This information should be presented in the form of stories (cf. Schank, in press).

The idea that learning takes place on a need-to-know basis is inherent in the case-based teaching architecture. Thus, there are two major issues in presenting cases for teaching: the construction and exploitation of a case base from which the cases can be drawn when they are needed and the creation of a situation that would cause a student to be interested in hearing about a relevant case. The case base must be indexed so that a description of a given situation can be used as a label to retrieve stories relevant to that situation. The juxtaposition of a case base and a situation that indexes the case base is the essence of case-based teaching.

Incidental Learning Architecture

Not everything is fun to learn. In fact, some things are terribly boring to learn. And when some member of a curriculum committee decides that such things must be mastered, often they will be taught in the easiest way possible as a list to be memorized or as a set of items to be tested on. But people regularly learn a variety of information that is quite dull without being completely bored by it. They do this by unintentionally learning that information. For example, most Americans know the 50 states that comprise the United States. They know them well enough so that it is a rare American who, when confronted with the word *Utah* or *Tennessee,* cannot identify it as a state. On the other hand, most Americans cannot name the 50 states. They may be able to name 47 but, in general, it is very difficult to remember each and every one. People do not have this problem with the alphabet. Most everyone can name every letter. The difference here is neither one of length of the list nor attributable to frequency of use. The explanation is quite simple: Everyone memorizes the alphabet, but almost no one is forced to memorize a list of the states.

It might seem that the natural conclusion from these observations is that memorization works, as indeed it does, if your goal is to make sure that the memorizer can recite a list back to you. On the other hand, if your goal is the acquisition of useful information, memorization is not a fruitful technique. Rather, we must ask the question, how did all these states come to be recognizable entities?

Much of what we know we have learned "in passing." That is, we have accumulated knowledge simply by the act of living and doing things that happened to include situations in which information comes up and was of use. We may know something about Iowa because (a) we read a book that took place there, (b) we knew someone from Iowa, (c) we played a game that involved learning something about Iowa, or (d) we like to watch college football. Our knowledge comes from experience with a subject and is thus scattered around memory stored with those experiences (Schank, 1982, in press). Thus, we may find it difficult to retrieve a set of facts about Iowa when asked to produce them, but we may find that information when we need it when it comes up in a natural way.

Much of what we learn in school involves presenting information in a manner that is quite different than the natural way (J. S. Brown, Collins, & Duguid, 1989). Suppose we expect students to learn lists of items or sets of facts. The method generally used is to tell them these facts and hope that the students remember them. But it is much more reasonable to expect students to remember facts that they gathered for some use or for some real purpose in which they were interested.

The task, then, is to design software that will present students with situations that are inherently interesting. We can use those situations to teach certain information by causing the interesting situations to present themselves only if the desired material is learned. Obviously this method can be misused. We could require students to recite the alphabet and then let them watch TV. This might work to a limited degree, but it is very important to keep in mind that the extent to which the material to be learned and the reward are really intertwined in a natural way strongly relates to holding a student's interest.

The architecture here, called the *incidental learning architecture,* involves the creation of inherently fun tasks to learn otherwise dull material. There are likely to be many subarchitectures involved here. In other words, there are many methods of doing this. Each method would serve as a standard software tool into which relevant material could be put to create new learning situations.

In other words, the trick to exploiting incidental learning is to find things that are inherently fun to do on a computer (e.g., any good video game). The next part is trickier. What the student naturally wants to learn in the video game ought to be worth learning. The trick then is to change the skills to be learned from eye–hand coordination tasks to content-based tasks, for which one needs to know real information in order to accomplish one's goal on the computer. This will work very well if there is a natural correlation between the content-based tasks and the inherently fun reason for doing the task. One should not have to learn verb conjugations in order to get to throw darts in a computer game.

Cascaded Problem Sets

We tend to teach problem solving by giving students rules to apply and problematic situations in which those rules might be useful. Then, we increase the complexity of the problem and hope that the student can adapt. Often such material is presented in a workbook format with text that is doing the teaching and problems that test what is supposed to be learned.

This format for teaching is ubiquitous in the schools and in training situations in business. It has many inherent problems. First, it is usually required that students do every example, as each one is a prerequisite to the next. Second, there is no remediation if a student fails to grasp an idea in time for the problem that tests it. Third, the material often proceeds too slowly for the taste of the student, often the textual material is dull and irrelevant to the problems to be solved. All in all, this is a dull way to teach, and most students object.

Given the opportunity, especially when these books are not used in a classroom under the watchful eye of a teacher, students will simply go to the back of these books and attempt to answer the last, and therefore most difficult problem, if indeed these problems have built on each other. If they fail to answer that problem, they will read as much of the book as they need to in order to answer it.

In general, school is much too problem oriented, much too concerned with getting students to learn the one and only way to do things that would enable them to respond with properly certified correct answers. But there can be no doubt that under certain circumstances there are right answers, and skills needed to determine those answers that are necessary to obtain. When this is the case, the teaching problem is to present the necessary information to students as they need to know it, when they want it, and in response to what they have been doing so far. This is the point of the *cascaded problem sets* architecture.

In this architecture, the idea is to build a problem space whereby each problem relates to each other problem with respect to the extra layer of complexity of a certain kind that it entails. In other words, "if you can't solve Problem A, you certainly can't solve Problem B" means that B is logically above A. Between A and B would be some information that B entails that A does not. Below A would be something simpler than A that perhaps does not entail the knowledge common to A and B. As students have trouble with one problem, they move down the cascade of problems by learning about the issues that one would need to know to solve the problem they were having trouble with. The tutor's task is to determine how to guide the student down the path of problems that address the issues the student is prepared to deal with and needs to deal with in order to move back up the cascade.

Teaching, when it occurs, occurs in response to students' request to help them with something they do not understand. In this way, learning is student-directed and relevant to a particular student's need. This architecture would be of use when there are skills to learn that depend on skills that should have been learned before.

In order to exploit this architecture, it is necessary to build libraries of cascades and to determine the content-based connections that relate one problem to another (Burton, J. S. Brown, & Fischer, 1984; VanLehn & J. S. Brown, 1980; White, 1984; White & Frederiksen, 1990). To do this, a problem must be decomposed into its constituent parts. Each constituent would itself be a problem, and it too would have constituent parts. This cascade of problems would go down until a basic level of knowledge that, it would be assumed, any user of the cascade would have. For example, at the bottom of a cascade of algebra problems would be basic arithmetic.

Determining how one problem is related to another is the major issue in the construction of the cascade. Exploiting the cascaded problem sets architecture means analyzing a domain of knowledge well enough so that the cascade can be built and the connections can be set up.

Directed Exploration of Video Database Connections

The best teachers we have ever had were the best storytellers. We remember physics teachers telling their stories vividly and with great drama, history teachers telling good stories from history, and english teachers telling good stories about former student's writing problems or about the lives of famous authors. We often have trouble remembering anything else but the stories, the material we were quizzed on has long since vanished.

The reason that we remember the stories teachers tell is that human memory is set up to retrieve and tell stories, as well as to capture the stories that others tell (Schank, in press). The story is a unit of memory. Furthermore, good stories contain good images, novel ideas, or particularly poignant passages that enable our memories to create indices that make retrieval of these stories easier. Storytelling depends on being reminded of a good story to tell. And, reminding depends on having labeled the stories we have heard or have created well enough so that when those labels appear naturally in the course of a day, we can use them to find relevant stories.

A good teacher tells good stories. He or she tells them in a way that we can remember them, and at a point when we might be able to understand their significance. A good teacher has a library of stories to tell and has knowledge, often implicit, of when the right time to tell a story might be.

It follows then, that the creation of effective software for teaching ought to allow the teachers who would be part of the process of building that software to capture their own stories and have the programs that are built tell those stories when appropriate, according to the rules that the teacher implicitly knows.

It is common to teach by giving assignments to students. Students are asked to write reports by going to the library and researching a topic (the research method). The concept here is a good one. Let students discover what they might find interesting and then have them organize what they find into a coherent report. The problem with this method of teaching, as discussed earlier, is that students are not always allowed sufficient freedom to insure that the task is always interesting to them. Often they are told what books to read, or, alternatively, they cannot easily find what might be of interest to them. Also, the topics they pursue are frequently assigned to them, regardless of their interests, as if it were only the content being taught and not the acts of discovery and reporting as well.

This instruction method has its flaws, but it can be used quite effectively if it is employed correctly. Even more effective, however, would be the creation of video databases that were easily explorable. There is a tremendous amount of information available on video. Go to any video rental store and see what is commercially available to the general public. But beyond movies and workout tapes there exists, in the archives of television networks for example, a tremendous amount of footage of the important world news events in the last 30 years, important leaders in every field being interviewed, studies of animals in the wild, instructional material, and so on.

Imagine if that material was available to any student who wished to view it. Children who have grown up on television are more receptive to video than to print. And whereas print authors can analyze a subject very effectively, nothing compares to seeing it for yourself.

The problem here is one of organization. It does not help anybody to make thousands of hours of video available without an organization scheme that allows users to find what they want instantly. Furthermore, such a system is of no use if it does not tell its users what information is available, information that they would not have known to look for at precisely the moment when they needed it.

Video databases would be valuable things if they were indexed in such a way as to present information organized by the content of that information, always pointing the way toward other available information that is similar in content, where similar is defined in a variety of different ways. To do this effectively, the concept of a show of an hour-long set of video clips, is irrelevant. It is the clips that comprise such shows that must be organized so that material on "man's inhumanity to man," for example, shown in thousands of shows or news programs, would be available by searching on various subtopics within that theme.

Any scheme of this kind would cause any one video clip to have multiple indices, because it could relate to many possible points. It is clear that the *indexing problem* (cf. Riesbeck & Schank, 1989) would cause the creation of such video databases to be a tremendous undertaking. Nevertheless, the rewards of such an undertaking, if properly executed, would be tremendous, not only for interesting students in finding out about the world but also as an archive of past history that has a variety of useful purposes.

The problem here is one of organization of information. After all, standard book libraries are essentially giant explorable databases. Indeed, any textbook is itself a set of information organized in a left-to-right fashion or by a key concept index in the back of the book.

Both of these schemes fail to work effectively because any good piece of information causes questions to come to mind that make the reader want to know more (Schank, 1986). There are many possible questions that an individual might ask after hearing someone say something or reading something. The idea behind this architecture is that it should be possible to ask these questions and have the answer be the very next piece of information presented by the system. This is possible if the information that relates to that question has been determined and collected and if it has been indexed in a such way that it can found or run into at the right point.

The key problem then is anticipating how information might provoke questions and making sure that those questions have been answered and that the answers are readily available. To do this requires having indexed the answers so the questions and indexes are one and the same and users can readily understand what questions can be asked.

Simulation-Based Learning By Doing

There has been one terrific piece of educational software that was created specifically for education purposes. It was created for a situation in which good education and training was critically important, when the obvious teaching method of trial and error was inappropriate because of the cost and danger involved, and when, most important, the money was available for the creation of the special software (and hardware) that was required. That piece of software is the air flight simulator, a device that is massively expensive, extremely effective, and better than having pilots practice takeoffs, landings, and emergency condition situations with actual 747s.

We all know that the best way to learn to do something is to try doing it.

This works very well when one is trying to learn to hit a baseball or drive a car. But when experimentation is impossible or very costly, we tend to devise other methods for training, ones that involve learning the theory of a subject, a great deal of classroom time, and assigned readings. It would be nice if software were available that served to simulate a great many situations that might be educationally relevant. To put this another way, it is possible to begin to regard training as primarily the testing of hypotheses by students who are learning by doing.

People love to explore, especially if the consequences of that exploration are unlikely to be dangerous in any way. Children are natural explorers (cf. Montessori, 1966). Teaching can take advantage of this inherent drive to discover how to do things by oneself. Unfortunately, most efforts in this regard have built-in perils. Take, for example, the chemistry set. Many middle-class children in the 1950s received chemistry sets as gifts. They excitedly mixed chemicals, hoping something fantastic would happen. What exactly did they expect? Obviously, most wanted to make an explosion of some kind. Now the designers of these sets knew that, so they made sure that explosions were impossible and thus defeated their own purpose.

A simulation could have solved all these problems. If experimentation is dangerous or impossible or very expensive or impractical in real life, it might be perfectly feasible with a good computer simulation. Let children make simulated explosions if they have sufficiently understood the chemical principles involved, and children will be beating down the door to get an electronic chemistry set and start learning chemistry.

A further advantage of creating simulations is that they provide an environment where a student can feel more comfortable making mistakes (and learning from them). In a "private," one-on-one interaction, students are freed from the social onus of failure, allowing their natural exploratory inclinations to emerge (Papert, 1980). Students' beliefs that making mistakes means that they are "dumb" is very discouraging, especially to weaker students (J. S. Brown & Burton, 1978).

As we previously argued (Schank & Farell, 1988), the task for software designers is to build more simulators and to begin to look at the education world in terms of the kinds of tasks and situations that can be simulated. If you want to learn how to deal with a client, use the client simulator. If you want to learn about investments, use the market simulator. If you want to learn about how to survive in the jungle, use the jungle simulator. If you want to learn about Napoleon, talk with the Napoleon simulator.

Obviously such simulations are very difficult to build. But it is possible. After all, some already exist. One very important role of computers in education is to create effective, realistic, simulators from which real-life lessons can be learned by play and experimentation. If you want to learn to throw a football, drive a car, build a mousetrap, design a building, teach students, cook three star cuisine, or be a management consultant, you must simply learn to do it by doing it. In many cases, parents are often the teachers, sitting nervously in the passenger seat while a teenager tries out the driver's seat for the first time. We hand a child a ball and teach him or her to throw, but the risk is much less. If the child throws poorly, he or she simply tries again.

However, for a class of tasks, instead of allowing students to learn by doing, we create instruction courses that tell students about the task in a theoretical way, without concentrating on doing the task. To put this another way, when the apprenticeship method will not work all that easily, we try lecturing (the sponge method).

The teaching architecture implied by all this is to change every possible skill into a learn-by-doing situation. To do this, one must create simulations that effectively mimic the real-life situation so well it prepares the student for real life situations without having to be in them. The air flight simulator has been effectively employed in pilot training because there was money available to build such simulators, and because the risk of injury in training by the more normal learn-by-doing method was quite high.

Simulations of all kinds can be built. What is required is to understand the situation to be simulated well enough so that the simulations are accurate portrayals. This can mean, in the case of simulations of people-to-people interactions, having to create complex models of human institutions, planning, and emotional behavior. These simulations would have to be built in such a way as to have roles available that students could play within the simulation. Thus, to learn to be a loan officer in a bank, students would play the role of loan officer in the simulation, trying out new situations that would cause them to have simulated experiences analogous to those they might encounter in the real world.

In order to build such learn-by-doing environments properly at least four basic components are necessary: (a) the simulator itself; (b) a teaching program that helps students through the simulator and can discuss issues with them; (c) a language understanding program that can understand questions students might ask; and (d) a storytelling program that would tell stories that represent the experience of experts in situations encountered by students in the simulation. These stories would be activated by states of the simulator and told by the teaching program when appropriate.

This teaching architecture is critical when the subject matter to be learned is one that is really experiential at heart. Much of natural learning is the accumulation of experience (J. S. Brown, Collins, & Duguid, 1989). Schools have trouble allowing children to learn from experience both because their individual experiences are so different and because the classroom situation does not allow for much actual experience to occur. Nevertheless, experience is the best teacher.

Responsive Questions

We can, of course, be our own teachers. Sometimes we have a good idea or a problem or just an issue that we would like to discuss. When we find someone to talk to, that person may not be the best listener in the world. They might want to tell us what to do rather than just listen to us. A good listener is also a good provider of questions. People who are creative know how to provide those questions to themselves (Binet, 1909; Collins & J. S. Brown, 1988; Collins et al., 1989; Collins & Smith, 1982). They know how to both have an idea and constructively criticize it at the same time, a process called *knowledge worrying* (A. L. Brown, 1985; A. L. Brown & Campione, 1981; A. L. Brown & Palincsar, 1989).

The *responsive questions* teaching architecture is intended to provide good questions to someone who is thinking about an idea or a problem. This is a very important part of teaching because it teaches, among other things, self-reliance on one's own ideas. In the classroom such teaching is virtually impossible to find because the 1-on-30 nature of classroom instruction seriously inhibits a teacher from following the reasoning of any one student too carefully.

The problem is to create software that has good questions and asks them at the appropriate time. The questions need not be too specific. The idea is simply to know how to provide a thought-provoking question rather than a thought-stifling one. Such a program would be quite useful to talk to even if it never understood what you were saying (Kass, 1990).

A PERSPECTIVE ON COMPUTERS IN EDUCATION

Recently a movie about an innovative and exciting teacher with a unique teaching style made its appearance, it is called *The Dead Poet's Society*. One interesting thing about this movie is how easily the Hollywood script writers imagined a character who was a good teacher. And this character was a good teacher. The students were excited to be in his classroom. In an era when good teaching appears to be hard to find, how is it that it is obvious to so many writers, actors, and the audience, what constitutes good teaching?

One is inclined to wonder why teachers cannot behave and teach the way the main character did when he played the teacher's role. One reason is supplied by the movie. At the end the teacher gets fired because he is blamed for the consequences of allowing his students to think freely about their own lives. Another reason, obvious to any teacher, is that this teacher would have been fired in any case because he probably was not teaching the required curriculum.

Many teachers know how to be interesting and are interesting. The problem is that school systems do not know how to make learning interesting.

What is the right subject matter to teach? The classics? Trigonometry? History? Computer literacy? It is easy to suggest that all these subjects ought to be taught, but they may not be the right subjects at all. What makes any one of these subjects necessary parts of the curriculum? How do we determine what is important to know? The problem of what to teach bears looking at, but one cannot look at this problem without also considering how to teach. What and how to teach are intimately entwined.

Ordinarily, subject matter would be selected before starting a discussion of how to teach that subject matter. But we do not live in ordinary times, educationally speaking. We discuss one example of this next.

Joshua Reading: A Case Study

My son Joshua has always had a difficult time in school because he has consistently found it boring. For example, he complained about having to do hundreds of arithmetic examples long after he had mastered whatever principle those examples were intended to illustrate. He knew how to read by the time he was 3 years old, and one consequence of this was that when reading was the subject matter in school he usually disliked it. Now this is an important point. Why should being good at reading cause one to dislike the subject when it is encountered in school? One obvious answer here is that perhaps the teachers were unable to cope with his being ahead of the rest of the class; thus, he was always doing work that was beneath him. "You shouldn't have pushed him," is a common response here.

In actuality, this was not exactly the problem. In Joshua's fourth grade class, students were grouped by ability in particular subject areas. Consequently, Joshua was in the advanced reading group with two other children. Nevertheless, Joshua was in trouble. I received quite a few notes from the teacher explaining that Joshua was not doing his work; eventually I was called in for a conference with the teacher. She was an energetic young woman, quite bright, and very motivated. She was genuinely concerned about Joshua. I asked to look at the work that Joshua was refusing to do.

Joshua's reading textbook looked like a giant SAT test. There were paragraphs to read and questions to answer about the paragraphs. There were vocabulary questions; there were multiple-choice questions about syllables, main ideas, grammar, and a range of other subjects. In short, there was a great deal of tedium and very little actual reading. I could see why Joshua was bored and why he had refused to do this. In fact, the teacher could see this too. She said that it looked boring to her as well.

I asked if I could make a radical suggestion, and she said she would listen to anything. I proposed that Joshua be given a book to read. The teacher agreed.

Weeks went by and reports from school were encouraging. Joshua liked to read after all. He was being challenged by the books that the teacher had selected. All his school problems, at least in reading, cleared up immediately.

Then I got a call from the teacher. She said that although this new strategy had worked very well for Joshua, the other children in the advanced reading group had not liked it at all. They had gotten into the advanced reading group, after all, by being very good at taking endless multiple-choice tests. That is what they knew, and they wanted to go back to it. Their parents had complained.

At least the teacher did not get fired. Innovation in the curriculum is not easy to accomplish. We revere the way things have been done. We think that what has been taught to us, ought to be taught to our children as well. Schools tend to reinforce the status quo even when everyone agrees that the status quo is not working.

The Role of Computers in Education

Where do computers fit? Their primary role in education will probably not turn out to be what most people now anticipate. Of course, computers can teach computing; they should be good at teaching various subjects in a way that is better than what now is done in those subjects. But, their major role when all is said and done will be to have served as an instrument of change. Surely they should not reinforce the status quo. Unfortunately to date, they have simply served to make those who clamor for change more hopeful that they are getting it.

One problem is that most people's reaction to the idea of getting computers into the schools is that it would be good for children to learn about computers. Everyone believes that children must learn about computers or learn to use them.

Two things seem clear when it comes to the public's evaluations of computers in the schools. First, nearly every parent of a school-age child is in favor of the school that their child attends having lots of computers. It is the mark of an advanced school system. Go to a town and ask about the quality of the school system. The computers that school has just purchased are immediately cited as an example of how progressive or rich or modern or high quality their school is. But try asking what the kids do with these machines. Parents do not know what the computers are used for and feel uneasy about the possibility that children may be learning less mathematics as a result. After all, the computers that they are familiar with are calculators, and the more you use your calculator, the less you use your head. Gone are those important math skills, like adding, because computers do it better.

In any case, that seems to be what adults feel about what children do with computers, but what is truly the case?

Actually, children do go to the computer room to learn something worth learning. Children want to learn, but unfortunately, there are not that many things one can learn on a computer. One cannot learn a subject if there is no software available to teach it.

BUTTON CONTROL: MAKING LEARNING ACTIVE

When personal computers became available and software was needed for them, especially because children were one primary target of the computer manufacturers' sales pitch, educational software began to be produced en masse. In general, there were two classes of people interested in building such software. The first class were the computer types who knew there would be a need for educational software, who knew how to build computer games, and who figured that anybody knew what there was to know about education. So they built games where children could shoot down the verb as it flew by and other such educational "masterpieces."

Educators, on the other hand, knowing nothing about computers, built programs that strongly resembled books. The child was asked to read a page, press "a," "b," "c," or "d" as an answer to the question asked about the passage and press the return key for the next page (i.e., the page-turning architecture).

There are an increasing number of educators and others who claim that computers have no place in the schools. Actually, most of the people who are against computers in the schools are against them because of the software that already exists. They have seen the reality and they do not like it. In fact, it seems quite clear that computers have no place in education if you look at the preponderance of misguided eduational software that is on the market.

One reason for this is that computers often make adults and children feel that they lack control. Control is a very important part of learning. To illustrate this, let us look at TV as an analogy. When people watch TV these days, they often sit with their hand on the remote control. With the advent of cable television, the "clicker" has changed the way people watch television. The essentially passive situation of watching what the networks offer has been replaced by a somewhat more active situation. Viewers can express their contempt or boredom by switching channels. In addition, they can exert some control over other viewers in the same room by switching channels at times when others object. Who controls the clicker is a real issue in some families.

What is especially interesting about the behavior of TV viewers with clickers is that they have converted a passive medium in which they have virtually no control to one in which they have a real control over other viewers in the room and a perceived control over what they themselves are watching. With 40 or 50 channels to choose from, there are some real choices to be made, and a viewer can be considerably less passive about what he or she watches.

Education is also an essentially passive activity, at least in its current formal implementation in our society. Teachers lecture, students take notes (i.e., the sponge method). In lecture classes, students are able to exert considerably less control than they could if they were at home watching TV. Their choices in a lecture hall are simply whether or not to be there. They have no other way of controlling what is presented to them.

In a class where questions are permitted, more control is possible, of course, but the difference is not that great. Lecturers rarely change much of their lecture as a result of students questions, so even if students get their question answered the lecture is likely to go on in the same way it would have proceeded had they not asked their question.

In a class that is more interactive, the perceived control may be greater. Even in a high school discussion class the teacher has a lesson plan, an agenda of what material has to be covered, and the lesson marches on, although there may be digressions. Students do indeed have more control in such classes, but like their counterpart with the clicker, although the whole family watches TV, no one person can exercise too much control without inviting the objections of the other viewers. It is not possible for students in a class to exercise the control that they would like to have, because the teacher cannot respond to divergent requests at the same time, even if the curriculum would allow it. So school is, in a sense, worse than television, at least with respect to control of what one is viewing.

How important, then, is control in education? For some students, it could make the difference between information that captures their attention and responds to their interests and lecturing that causes doodling and note passing.

What kind of control could be given to students? To consider this issue, let us return to our TV viewer. What kinds of buttons would a TV viewer choose if they could have any buttons that they wanted? Right now, with the exception of unimportant buttons like "volume control," the TV viewer has essentially two: "change channel" and "mute." The channel-changing button can be viewed in more conversational terms as a button that says to the TV, "give me something else to look at please." (The mute button has to do with issues not inherent to information viewing—e.g., commercial interruptions or having to answer the phone—so we can ignore it here.) The question then is: What other things would a viewer like to say to his or her TV besides "give me something else"?

It is not really necessary to study the viewing habits of TV watchers or the desires of students in a lecture hall in order to answer this question, because this is not a question about TV or about passive students. This is really a question about active learning, or, to put this another way, this is a question about conversation with someone who will not listen to your stories but will listen to your requests. This point is worthy of further elaboration.

Recently, we have been working on the problem of storytelling and conversation (Schank, in press). Most conversations are really mutual storytelling exchanges. One person tells a story to a friend, and the friend responds with a story of his or her own. Not all stories are interesting. In fact, not all stories look like stories. Here we define a *story* as something that has been thought about previously and prepared as a set of information to be imparted under certain conditions. People wind up telling the same story over and over again and are, in some sense, looking to tell their stories at appropriate points in a conversation.

Sometimes, but not always, people actually listen to the stories they are being told. They do this well enough to determine which of their favorite old stories is relevant to respond with. They also listen in order to learn from the stories they are hearing. Although this is rare enough in most conversations, it is the norm when the listener is trying to learn something from the stories being told. So, our important issue then is—what kinds of things do listeners say when their goal is to learn from the stories they are hearing?

But this is the deep form of the question. Actually, we are not so concerned with what listeners say as with what they mean. If listeners are bored with the speaker's story, for example, they are unlikely to say that they are bored, as it is rather impolite to do so. People often do not say what they mean because they are concerned that someone will take offense. Of course, no such problem would exist with a computer, so we really want to know what people would say if no one could possibly take offense in a situation where information is being presented to them. Or, put another way, what buttons should we put on the ultimate clicker?

In answering this question, we imagine a machine that can respond to any button we select in the way we would intend for that button to be interpreted. In other words, there is a difference between what the label of a button might be and a viewer's real intent in pushing it. Buttons can be ambiguous with respect to intent. Thus, a machine that could both correctly interpret a viewer's intent in pushing a particular button and be able to respond with what it interpreted that the viewer would like to next see or hear would be an intelligent machine indeed. No such machine exists at the moment. However, answering the question of what buttons should, in principle, be available, will allow us to begin building this machine. Intelligent storytelling or interactive teaching depends on giving control to the listener. Our question is: What controls does the listener need?

TEACHING BUTTONS: AN ARTIFICIAL INTELLIGENCE (AI) PERSPECTIVE

What if there were a "boring" button? Imagine if every remote control contained a boring button. What would happen when it was pressed? Clearly, the answer today would be "nothing." What would happen in the ideal situation? The TV would be smart enough to correctly interpret the intent of this button. What does it mean to correctly interpret a button? Certainly, the issue of correct interpretation of the change channel button never comes up. But a smart TV would have to have three important properties. First, it would have to know what exactly the viewer's problem was that he or she expressed as boring. Second, it would have to be able, on analysis of the viewer's state, to determine what new information it had that might address the problem. Third, it would have to determine what information was available that best fit the type of information the viewer wanted, find that information, and present it.

This kind of intelligent TV would not be *intelligent* in the standard sense of the word. Much of the AI literature is full of problems that address the issue of determining sufficient information about the input—natural language analysis, student modeling, and such—to allow a machine to respond effectively. Here, the problem is determining what a machine ought to know and ought to say in order to respond effectively to what the user wants. In other words, the intelligent TV would be a very different kind of AI program. It would be a system with many stories to tell equipped with methods of determining which one it should tell at any given moment. Its primary job would be to determine the overall context and respond accordingly. It would always be trying to find something relevant to present. Thus, it is limited by what is possible for it to present.

The boring button allows users to complain, but only partially makes clear the nature of their complaint. Determining what their complaint actually is involves knowing what they were watching and having tracked what path they were pursuing up until the boring button was clicked. Furthermore, if after being presented with new information following a press of the boring button, the user again pressed the boring button, the system might need to respond differently the second time. In other words, the boring button means different things at different times depending on the context.

From an AI perspective, the problems here include tracking the behavior of both the user and the system to know what has been presented and why in order to determine what next to present. In other words, the intelligent TV tries to determine what its viewer would like to see based on the information provided to it by the viewer in much the same way as persons in a conversation endeavor to listen well enough to determine which of their stories to tell next.

The problem for us is to determine the set of buttons that is of the most use in an intelligent storytelling program (see Appendix for a rough example of such a set of buttons). My son Joshua would have liked a "move faster" button, for example. Most children would likely enjoy and use a "why" button or a "what do I do now" button. Determining the set of controls that can be given to computer users also determines the nature of the intelligence that needs to be put into computers in order to interpret them. The buttons we select will necessarily determine the kind of intelligent processes that an intelligent, controllable, teaching machine needs. In other words, the inputs will drive the processes needed to determine and present the output.

The issue in education is control. Who is in control? Now it is curriculum designers, lecturers, and workbook publishers. In many cases it ought to be the students. Most students want to learn. When they determine that their questions will be answered, they will keep asking questions. The real role of the teacher, computer or human, is to keep the environment interesting enough to prompt questions.

Can computers be successfully employed as a means of educating children? It is important to start designing computer software that makes sense from an educational point of view. The issue is not what is available but what could be available. The issue is not the quality of existing software, but the possibilities of what could be created for the schools if the best computer scientists, educators, and psychologists put their heads together.

ACKNOWLEDGMENTS

This research took place at The Institute for the Learning Sciences which is supported in part by the Defense Advanced Research Projects Agency, the Air Force Office of Scientific research, the Office of Naval Research, IBM, Ameritech, and Andersen Consulting. The Institute was established in 1989 with the support of The Arthur Andersen Worldwide Organization.

We thank Allan Collins for helpful discussion and comments on a draft of this article.

REFERENCES

Apple, M. (1979). *Ideology and curriculum.* Boston: Routledge & Kegan Paul.

Bain, W. M. (1986). *Case-based reasoning: A computer model of subjective assessment.* Unpublished doctoral thesis, Yale University, New Haven, CT.

Bereiter, C., & Scardamalia, M. (1985). Cognitive coping strategies and the problem of "inert knowledge." In S. Chipman, J. Segal, & P. Glaser (Eds.), *Thinking and learning skills: Research and open questions* (Vol. 2, pp. 65-80). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Binet, A. (1909). *Les idees modernes sur les infants* [Modern ideas on children]. Paris: Ernest Flammarion.

Bloom, B., & Broder, L. (1950). *Problem-solving processes of college students.* Chicago: University of Chicago Press.

Bransford, J. D. (1979). *Human cognition: Learning, understanding, and remembering.* Belmont, CA: Wadsworth.

Bransford, J. D., Franks, J. J., Vye, N. J., & Sherwood, R. D. (1989). New approaches to instruction: Because wisdom can't be told. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 470–497). Cambridge, England: Cambridge University Press.

Brown, A. L. (1985). Mental orthopedics: A conversation with Alfred Binet. In S. Chipman, J. Segal, & R. Glaser (Eds.), *Thinking and learning skills: Research and open questions* (Vol. 2, pp. 319-337). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Brown, A. L., & Campione, J. C. (1981). Inducing flexible thinking: A problem of access. In M. Feldman, J. P. Das, & N. O'Conner (Eds.), *Intelligence and learning* (pp. 515-529). New York: Plenum.

Brown, A. L., & Palincsar, A. S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 393-451). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Brown, J. S. (1985). Idea-amplifiers: New kinds of electronic learning. *Educational Horizons, 63,* 108-112.

Brown, J. S., & Burton, R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science, 2,* 155-192.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32-42.

Brown, J. S., & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science, 4,* 379-426.

Burton, R., Brown, J. S., & Fischer, G. (1984). Skiing as a model of instruction. In B. Rogoff & J. Lave (Eds.), *Everyday cognition: Its development in social context* (pp. 139-150). Cambridge, MA: Harvard University Press.

Carroll, J. B. (1968). On learning from being told. *Educational Psychologist, 5,* 4-10.

Collins, A., & Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 1-18). New York: Springer-Verlag.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Collins, A., & Smith, E. E. (1982). Teaching the process of reading comprehension. In D. K. Detterman & R. J. Sternberg (Eds.), *How much and how can intelligence be increased?* (pp. 173-185). Norwood, NJ: Ablex.

Frase, L. T., & Schwartz, B. J. (1975). Effect of question production and answering on prose recall. *Journal of Educational Psychology, 67,* 628-635.

Hammond, K. J. (1989). *Case-based planning: Viewing planning as a memory task.* Boston: Academic.

Kass, A. (1990, March). Are electronic tutors really what we want to build? In B. Woolf & E. Soloway (Co-chairs), *Knowledge-based environments for learning and teaching.* Symposium conducted at AAAI Spring Symposium Series, Stanford University.

Kolodner, J. L. (Ed.). (1988). *Proceedings of the case-based reasoning workshop.* Clearwater Beach, FL: Kaufmann.

Kolodner, J. L., & Simpson, R. L. (1989). The MEDIATOR: Analysis of an early case-based problem. *Cognitive Science, 13,* 507–549.

Lepper, M. R., & Greene, D. (1979). *The hidden costs of reward.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Malone, T. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive Science, 4,* 333–369.

Montessori, M. (1966). *The secret of childhood.* New York: Ballantine.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Riesbeck, C., & Schank, R. (1989). *Inside case-based reasoning.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Scardamalia, M., & Bereiter, C. (1991). Higher levels of agency for children in knowledge-building: A challenge for the design of new knowledge media. *The Journal of the Learning Sciences, 1,* XX–XX.

Schank, R. C. (1982). *Dynamic memory.* Cambridge, England: Cambridge University Press.

Schank, R. C. (1986). *Explanation patterns: Understanding mechanically and creatively.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Schank, R. C. (in press). *Tell me a story: A new look at real and artificial memory.* New York: Simon & Schuster.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Schank, R. C., & Edelson, D. J. (1990). A role for AI in education: Using technology to reshape education. *Journal of Artificial Intelligence in Education, 1,* 3–20.

Schank, R. C., & Farell, R. (1988). Creativity in education: A standard for computer-based teaching. *Machine-Mediated Learning, 2,* 175–194.

Schoenfeld, A. H. (1985). *Mathematical problem solving.* New York: Academic.

Smith, K., Johnson, D. W., & Johnson, R. T. (1981). Can conflict be constructive? Controversy versus concurrence seeking in learning groups. *Journal of Educational Psychology, 73,* 651–663.

Toulmin, S. (1958). *The uses of argument.* Cambridge, England: Cambridge University Press.

VanLehn, K., & Brown, J. S. (1980). Planning nets: A representation for formalizing analogies and semantic models for procedural skills. In R. E. Snow, P. A. Federico, & W. E. Montague (Eds.), *Aptitude learning and instruction, Vol. 2: Cognitive process analyses of learning and problem-solving* (pp. 95–137). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Whimbey, A., & Lochhead, J. (1978). *Problem-solving and comprehension: A short course in analytical reasoning.* Philadelphia: Franklin Institute Press.

White, B. Y. (1984). Designing computer games to help physics students understand Newton's laws for motion. *Cognition and Instruction, 1,* 69–108.

White, B. Y., & Frederiksen, J. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42,* 99–157.

APPENDIX
Teaching Buttons

Button Name	*Summary*
Group 1: Feelings	
Awesome	An expression of enjoyment or excitement.
Boring	Self-explanatory.
No way	A way of expressing disbelief.
Huh	Student does not understand what just happened or what was just asked of him or her.
Too hard	Student feels over his or her head; the material is just too difficult.
Group 2: Questions	
Why	Student wants an explanation of something that just happened or wants to know why an action should be performed.
How do I do that	Student needs an explanation of how a requested action is to be executed.
Now what	Student wants to know what to do next or wants to know what the next step in a task sequence is.
What's the point	A summary is desired or an indication that presentation of introductory material should stop.
History	Provides a recap of the progress of the lesson so far or provides the student with some background on how the current situation came to be.
Group 3: Control	
Change task	Student wants to do something different.
Backup	Student wants to backtrack to a previous point in the lesson.
Big picture	Student desires an overview of how the current lesson fits into the overall scheme.
More detail	Student is interested in the current point and wants more information.
Skip this	Student feels comfortable with the current task or topic and wants to move on.