In S. Vosniadou, E. Corte, R. Glaser + H. Mandl (Eds.)(1996) International Perspectives on the Design of Technology-Supported Learning Environments, Mahwah, NJ: Lawrence Erlbaum, pp. 347-361

17 Design Issues for Learning Environments

Allan Collins
Bolt Beranek and Newman Inc., Northwestern University

Instructional design theory comes with a belief in instructional delivery. The goal is to design an instructional system that transmits content and skills in a clear, well-structured, and efficient manner. The approach derives from the behaviorist and programmed instruction tradition of Thorndike, Skinner, and Gagne, but has assimilated aspects of cognitve research in recent years. The approach has been well summarized in two volumes edited by Reigeluth (1983, 1987).

The constructivist view of education, stemming from the work of Piaget, Dewey, and Vygotsky, argues that the goal of education is to help students construct their own understandings. In contrast to the instructional-delivery view, the constructivist view leads to an emphasis on learning rather than teaching, and on facilitative environments rather than instructional goals. A design theory for a constructivist approach to education looks very different from traditional instructional design theory. This chapter is a beginning attempt to develop such a constructivist design theory.

When designing a learning environment, computer based or not, there are a multitude of design decisions that must be made. Many of these design decisions are made unconsciously without any articulated view of the issues being addressed or the trade-offs involved. It would be better if these design decisions were consciously considered, rather than unconsciously made.

The perspective I take on design is to think of each decision in terms of its costs and benefits. From this perspective, the crucial questions are: What are the issues that must be addressed in designing learning environments? What are the cost–benefit trade-offs associated with each design issue? How should the costs and benefits be weighed? The costs and benefits relate to the effects on student learning and motivation, and to the costs in terms of time, money, and effort required to implement any aspect of a learning environment. My goal in this chapter is to raise a set of issues and some of the cost–benefit trade-offs that arise with respect to each issue. This cost–benefit approach does not lead to design prescriptions, as in instructional design theory, but rather to identifying important issues to be weighed.

The first issue to address in the design of any learning environment is what is called *authenticity* (Brown, Collins, & Duguid, 1989; Wiggins, 1989). The questions associated with authenticity are: What are the potential uses for the knowledge? How can a learning environment be created that reflects those possible uses? Too much of what we teach in school is taught because it has always been taught. We need to rethink what students should learn to live in the 21st century. For example, should we spend 12 years teaching students mathematical algorithms that computer tools can carry out for them? The goal of authenticity is to prepare students to do the kinds of complex tasks that occur in life. Much of what is learned in school is never used because it is often the wrong knowledge for the modern world; even when it is the right knowledge, people do not know how to apply it.

In previous work (Brown, Collins, & Duguid, 1989; Collins, 1991; Collins, Brown, & Newman, 1989; Collins, Hawkins, & Carver, 1991), I have proposed a "cognitive-apprenticeship" approach to designing learning environments. Before the invention of schooling, everything was taught by apprenticeship, where learning is situated in the context of work. It is the most natural way to learn. The basic method of apprenticeship involves modeling, coaching, and fading; that is, first showing apprentices what to do, next observing and helping them as they try to do it themselves, and then fading the help as they take on more responsibility. Cognitive apprenticeship attempts to apply this approach to teach thinking and problem solving. But unlike the kinds of skills taught with traditional apprenticeship, thinking is not visible. Hence, cognitive apprenticeship stresses the importance of techniques to make thinking visible, such as articulating and reflecting on cognitive processes.

This chapter attempts to extend the cognitive-apprenticeship framework to address a broad set of issues that arise in the design of learning environments. The first section on learning goals discusses central issues about what we should try to teach: Should we aim for memorization or thoughtfulness, whole tasks or component skills, breadth or depth of knowledge, and so on? The second section on learning contexts addresses cost–benefit trade-offs in the learning environments created by the designer: Does the context promote activity or passivity, incidental or direct learning, fun or seriousness, and so on? The third section discusses sequence in the learning activities: Should learning proceed from grounded to abstract, from structured to exploratory, from simple to complex, and so on? The final section on teaching methods addresses cost–benefit trade-offs associated with the cognitive-apprenticeship methods, such as modeling, coaching, articulation, and reflection.

LEARNING GOALS

The first set of trade-offs that designers need to address has to do with what students should learn. The trade-offs I address in this section are: memorization vs. thoughtfulness; whole tasks vs. component skills; breadth vs. depth of knowledge; diverse vs. uniform expertise; access vs. understanding; and cognitive vs. physical fidelity.

Memorization Versus Thoughtfulness

There is a tension throughout school between students memorizing things to do tasks fast and easily, such as memorizing the multiplication table, and learning to do things thoughtfully, such as solving complex problems. To the degree that one knows how to do something automatically (e.g., decoding), it can free the mind to be thoughtful about other things (e.g., the meaning of the text).

To illustrate the issue, in the 1930s, the superintendent in Manchester, New Hampshire, persuaded some of his teachers to give up teaching math algorithms for the first 5 years of school, and instead to focus on math discussion and estimation tasks (Benezet, 1991). When he visited schools, he would give students problems such as, "If half a stick is buried in mud, two-thirds of the rest is under water, and 1 foot is above water, how long is the stick?" Students who were taught in the traditional manner would start adding or multiplying the numbers given, whereas the students taught in the new manner would reason through the problem. He clearly had opted for thoughtfulness over memorization.

Some of the costs of memorization are evident in the examples: Memorization leads to inflexible use of the memorized knowledge and reliance on drill and practice, which is unmotivating for students. The benefits are that the memorized skills can be off-loaded to free the mind for thinking. The mastery of knowledge and skills by memorization is also empowering for students. Gaining automaticity is crucial if one is going to use a particular skill a lot. But if one will hardly ever use a skill (such as multiplying fractions), memorization is not worthwhile. When automaticity is appropriate, it is best gained by creating practice environments that reflect the uses of the skill in the real world.

Whole Tasks Versus Component Skills

There is a trade-off between having students perform whole tasks that require integration of a variety of skills vs. having students perform simplified tasks that focus on particular subskills. For example, one can have students practice sounding out different phonic patterns, or one can have them read Dr. Seuss books for enjoyment. There is a tendency in school to break everything into easy components (e.g., how to multiply fractions), but it is often difficult for students to tell what the components are good for. Much of school is like having students practice the forehand, backhand, and serve needed for tennis, without letting them know what the game is.

The costs of giving students whole tasks are that it is difficult to focus on particular weaknesses, it is difficult to manage the whole process at once, and there is always a chance of failure when the task is too complex. The benefits of whole tasks are that it is easy to see the point of the exercise, it is possible to practice the integrative skills that are necessary, and one is unlikely to develop strategies (as students do for component-skills tasks) that are counterproductive to the task as a whole.

It seems clear that focusing on subskills is sometimes very productive, but ideally this should occur when a weakness has been diagnosed. Scaffolding (Collins, Brown, & Newman, 1989; Palincsar & Brown, 1984) permits even weak students to accomplish whole tasks from the beginning. One strategy is to start by scaffolding students in whole tasks, and then going to component tasks when they seem appropriate.

Breadth Versus Depth of Knowledge

The issue is whether we want students to learn a little about a lot of things, as Hirsch (1987) argued in his plea for cultural literacy, or whether we want them to understand a few topics deeply. Our society tends to value specialists more than generalists, and yet schools are pressured to include more and more information in their curriculum.

The costs of breadth are that students do not get an authentic feel for any subject, and that a demand for breadth often gets turned into requirements that students learn particular things. The benefits of breadth include cultural literacy, which, as Hirsch argued, is critical for people to understand each other. Students are also exposed to many different ideas so that they can make knowledgeable choices about which interests to pursue. Finally, breadth allows students to make connections between many different disciplines, which can provide novel insights. A possible compromise between breadth and depth is to pursue a few topics in depth while broadly covering a wide variety of topics. Some students should become specialists and some should become generalists, and learning environments should support both goals.

Diverse Versus Uniform Expertise

Most schools attempt to ensure that all students learn the same thing. An alternative goal is for students to gain diverse expertise. This difference has profound effects on the organization of learning. For example, in a middle school in Rochester, New York, Carver (1990; Collins, Hawkins, & Carver, 1991) had eighth graders conduct research on different aspects of the city, such as the history, climate, culture, and government, in order to produce a HyperCard exhibit for the Rochester Museum and Science Center. Students specialized in different content areas and different tasks (e.g., producing text vs. graphics for the exhibit). The traditional school approach would have students read and discuss the same material on these topics.

One cost of diverse expertise is the loss of a community of shared knowledge, where students can discuss issues from similar backgrounds. Another cost is that teachers can no longer evaluate students in the same terms: whether they have learned particular content or skills. The benefits are that students can specialize in what interests them, and will feel pride of ownership in the knowledge and skills they have and that others do not have. It can also be viewed as a benefit that teachers cannot measure students on a simple metric, such as how much specific content they have gained, but rather must judge them in terms of their products and efforts (Collins, 1990). Our best examples of teaching (Lampert, 1986; Resnick, Bill, Lesgold, & Leer, 1991; Stigler & Perry, 1988) rely on uniform expertise, but the introduction of new technology and a constructivist pedagogy fosters a change to an emphasis on diverse expertise (A. Brown, 1992).

Access Versus Understanding

As we give students more powerful tools, understanding of the ideas and procedures that the tools accomplish for us is lost. For example, if we give students tools that fix the spelling and grammar in text, or that compute all the math algorithms in school, then knowledge of how to do such things will die out among students.

The cost of giving students access to powerful tools is that students will not understand how these tools work, and thus will not be able to evaluate the products derived from the tools. The benefit of giving students access to powerful tools is that they can get on with learning what they will need for the future, instead of learning spelling and algorithms. Furthermore, they will be vastly empowered by having tools that do for them what people are not so good at doing. History is replete with lost understandings. For example, how to grow crops and make clothes were once taught to practically everyone, and it seems inevitable that much of what we now teach in school will not be learned by most people in the future.

Cognitive Versus Physical Fidelity

As we create simulated environments, either on or off computers, a critical question emerges: What is the trade-off between preserving physical fidelity to the environment vs. preserving only cognitive fidelity? This trade-off is well illustrated by the difference between a simulation of the steam plant on board ships built by the navy, which preserves all the physical details—filling two large rooms and requiring a crew of eight people to operate—and the cognitive simulation in STEAMER (Stevens & Roberts, 1983), which shows the configuration of the entire system and its different subsystems, as well as the flow of water and steam inside the pipes. It is much easier to understand the system from working with the cognitive simulation, but much of the physical detail is lost.

The cost of stressing cognitive fidelity is that learners may not recognize particular situations in the real world because they look different than in the simulation. Another problem is that important mappings that are used for understanding a system may be lost: Any simulation that throws away a large portion of the mapping to reality risks throwing away some critical elements on which people rely. The benefit of stressing cognitive simulation is that it makes it possible to focus on salient aspects of the situation, so that students do not get lost in complexity. Moreover, cognitive simulations are much cheaper to build. It pays to start with cognitive fidelity so that students get the big picture, and then move to greater physical fidelity.

LEARNING CONTEXTS

There are several trade-offs that have to do with the learning contexts created for students. These include whether the learning is highly interactive or not, incidental or direct, fun or serious, natural or efficient, and whether or not the learner is in control.

Interactive Versus Active Versus Passive Learning

There is a difference between active learning and interactive learning that is often overlooked. It is the difference between being in a highly responsive environment, such as playing a video game, vs. being in a fairly nonresponsive environment, such as working with a drawing program or LOGO (Harel, 1990). Both are different from passive learning, such as listening to a lecture or watching a video, where the learner can easily tune out what is happening.

The costs and benefits of active learning vs. passive learning are probably well known, but the costs and benefits of interactive learning vs. active learning are less well known. The costs of high interactivity are a lack of thoughtfulness by the student because things move fast, and a lack of problem finding and construction by students because everything they do is responsive to some situation. The benefits of high interactivity are that students receive immediate feedback on the success of their actions, they find such environments extremely motivating, and they are very active trying out different skills and strategies. Clearly, there needs to be a mix of highly interactive and less interactive environments for learning. Less interactive environments foster thoughtfulness, whereas highly interactive environments foster automaticity.

Incidental Versus Direct Learning

When students are in a task environment, what they learn may be taught directly by the task itself or only incidentally to the task. For example, the computer game, Carmen San Diego, is designed to teach knowledge about geography incidentally to tracking down criminals, whereas a travel agent simulation program, where students find places to visit meeting different specifications (warm climate, inexpensive), would teach about the uses of geographical knowledge directly. It is possible to create very engaging tasks if one is willing to teach indirectly rather than directly.

The costs of incidental learning are subtle and have to do with authenticity. To the degree that one teaches indirectly, it is likely to promote the wrong lessons: In Carmen San Diego, the geographical facts are mostly useless (e.g., that they speak French in Cameroon) and are not integrated in any well-organized structure. Hence, the knowledge gained is not likely to be of much use when trying to do any task that requires geographical knowledge. But of course the benefit is that the task is likely to be engaging, therefore students will spend more time at it and perhaps learn more geography. My own preference is to create engaging tasks that reflect the uses of the knowledge to be learned, and let any facts and concepts be learned incidentally.

Fun Versus Serious Learning

There is a tendency to think that it is good for learning to be fun, but there is a downside. The costs are that students do not take what they are learning seriously, and so may not remember it. They also do not learn to force themselves to do difficult tasks. They come to think that all learning should be fun, but unfortunately life is not like that. The benefits are that you reach more students, and they will spend extra time and effort. Furthermore, the repetitive drill and difficult tasks in school manage to turn off many students to education generally.

My own view is that it is best to engage students by creating meaningful tasks not by creating fun environments. An example is the project in Mississippi where African-American students collected oral histories from adults who lived through the civil rights struggles of the 1960s, which they published as a book. This was a serious task, but it was as engaging as any fun task.

Natural Versus Efficient Learning

Most of the natural ways we learn things are inefficient, and so there is always a tendency to try to design more efficient learning environments. For example, the way we first learn language in the home is different from the more efficient ways we try to teach adults a second language. The learning children do when they invent arithmetic algorithms is different from the learning of the standard algorithms in school.

The cost of naturalness is simply its inefficiency: It takes children years to learn to speak their language. People also do not naturally learn the most effective ways to do things, as with arithmetic. The benefit of natural learning is that it is functional: Such learning enables people to achieve their goals so that the success rate is high. Learners do not learn the kinds of counterproductive strategies that Schoenfeld (1985) described for school math learning. Sacrificing naturalness is probably fine as long as we do not sacrifice functionality for the learner.

Learner Control Versus Computer or Teacher Control

There is a trade-off between putting the learner in control of his or her own learning vs. keeping control by the teacher or computer. Exploratory environments (e.g., Physics Explorer) and tool-based projects (e.g., the one in Rochester, New York) largely give control to students, whereas intelligent tutoring systems such as the LISP, geometry, and algebra tutors built by Anderson and his colleagues (e.g., Anderson, Boyle, & Reiser, 1985) keep rather tight control over what the student can do.

The cost of giving learners control is that most lack knowledge about the structure of the domain, about how to learn effectively, and about what they know vs. what they do not know. Hence, they make poor educational choices for themselves. But the benefit of giving learners control is that they can study what is most interesting and challenging to them. Furthermore, control over their own learning is motivating to many students. One strategy is to give students control over everything but pedagogical decisions; another is to give students information to help them make good pedagogical decisions (Frederiksen & White, 1990).

SEQUENCE

Because a learning environment changes as a person interacts with it, one way to treat some of the trade-offs is sequentially. I propose that the trade-offs between grounded vs. abstract learning, structured vs. exploratory learning, systematic vs. diverse learning, and simple vs. complex learning be treated in this way.

Grounded Versus Abstract Learning

Learning contexts can mimic the situations in which the knowledge is likely to be used or they can be abstracted from particular situations. For example, in order to teach arithmetic we can put students in the context of running a bank or building

(Cuban, 1984), which are grounded in particular situations. Alternatively, we can teach students abstract algorithms that can be used in any context.

The costs of grounded learning derive from the fact that students' knowledge is tied to particular situations, and so they neither learn a general framework nor learn how to apply their learning to new situations. The benefits of grounded learning are that students see the point of what they are learning, and learn at least one way to use their knowledge. Furthermore, it is difficult to remember abstractions if they are not grounded in situations that are memorable.

Currently, mathematics education starts with abstract algorithms, and then teaches students how to apply these abstractions in particular situations, through story problems. We have argued elsewhere (Brown, Collins, & Duguid, 1989) that this is backward. Students should first learn knowledge and skills in context, and by experiencing multiple contexts they should learn to generalize their knowledge.

Structured Versus Exploratory Learning

Highly structured learning environments keep students engaged in activities that can lead to learning. For example, the LISP, geometry, and algebra tutors built by Anderson et al. (Anderson, Boyle, & Reiser, 1985) provide immediate feedback and correction in response to students' mistakes, and thereby keep students from going off the correct solution path. Other systems, such as Physics Explorer and Interactive Physics allow students much more flexibility to explore and even play, although Physics Explorer does allow teachers to set up structured exercises for students.

The costs of structured learning environments are that students do not learn to find their own problems, and they do not learn to explore productively. The benefits of structured environments are that students do not end up floundering or randomly playing, and they are not as likely to get turned off by failure. Ideally, students would start out in highly structured environments and, as they master the skills of the domain, move to less and less structured environments.

Systematic Versus Diverse Learning

The problems and tasks posed to students can vary in systematic ways or in more diverse ways, as they do in life. For example, in mathematics, one can give students a whole series of distance, rate, and time problems to solve, or one can have a mixture of many different kinds of problems.

One cost of giving students problems that vary systematically is that they will learn ad hoc strategies for solving the problems, which do not apply in other settings. Another cost is that they will not learn to figure out when a particular solution method or strategy is appropriate. The benefit of systematic variation is that induction is much easier, and so learning is much more efficient. Schoenfeld's (1985) strategy in teaching problem solving is to start with systematic variation and move to more and more diverse problems.

Simple Versus Complex Learning

There has been a tendency in education to simplify problems and tasks so that all students can succeed. For example, we give students *Dick and Jane* to read, rather than books like *The Hobbit*. The cost of simplification is oversimplification for many students: The tasks often become boring and meaningless. The benefits of simplification are that more students are likely to succeed, and thus it is possible to focus on important prerequisites.

In general, one wants to proceed from the simple to complex, but ideally one should start at the optimum complexity for each student. This may mean doing some simple inquiry or assessment beforehand to determine where to start. Scaffolding (Palincsar & Brown, 1984) is designed to get students through more complex tasks with just as much support as they need, but no more.

TEACHING METHODS

There are a set of teaching methods associated with cognitive apprenticeship (Collins, Brown, & Newman, 1989; Collins, 1991; Collins & Brown, 1988) that have both advantages and disadvantages. The methods I focus on here are modeling, scaffolding, coaching, articulation, and reflection. These are discussed in more detail in the earlier papers.

Modeling

Modeling processes for students can enhance their understanding dramatically. There are two kinds of modeling that are critical to consider in the design of learning environments (Collins, 1991): (a) modeling the physical processes underlying phenomena we want students to understand, and (b) modeling the thought processes underlying expert performance. For example, in the Quest system (White & Frederiksen, 1990), the system can model how electricity flows in different circuits, and how an expert troubleshooter would locate a fault in different circuits.

The costs of modeling are that it is a passive and often boring activity for students. The benefits are that they can see normally invisible processes, and they can begin to integrate *what* happens with *why* it happens. Modeling is potentially very valuable, but it seems best to model early in the learning process and involve the learner as much as possible.

Scaffolding

Scaffolding is the support given to students as they carry out a task (Collins, Brown, & Newman, 1989; Palincsar & Brown, 1984). It can come in many different forms: the short skis that enable people to learn to ski much faster (Burton, Brown & Fisher, 1984), the cue cards that Bereiter and Scardamalia (1987) give students to prompt them as they plan to write, and the hints that Palincsar and Brown (1984) and Lesgold, Lajoie, Logan, and Eggan's (1990) Sherlock system provide students as they carry out a task.

The cost of scaffolding is that it is a crutch that students know they can fall back on, and so they may become dependent on it. The benefits of scaffolding are that it helps students accomplish difficult tasks, providing focused help at critical times and only as much as needed. In designing learning environments, it is in fact easier to provide scaffolding than to provide the kind of coaching described next. Ideally, the scaffolding would be faded as students become more expert.

Coaching

Coaching involves a whole range of activities: choosing tasks, modeling how to do them, providing hints and scaffolding, diagnosing problems and giving feedback, challenging and offering encouragement, and structuring how to do things. For example, Heath (1991) described how a little league baseball coach gets students to view mistakes as learning experiences; Lepper, Aspinwall, Mumme, and Chabay (1990) described how math tutors challenge students to get them to try difficult problems and not be afraid of failing. The most elaborate computer coach to date is the coach for the game *How the West Was Won*, built by Burton and Brown (1982). The coach diagnoses the patterns of play students are following, and then makes suggestions at opportune moments as to how the students might improve their game.

One cost of coaching has to do with the dangers of misdiagnosis, which is likely with computer coaches because of their limited bandwidth for viewing student behavior. To the degree that the diagnosis is shallow, as in the Anderson et al. (1985) tutors, the likelihood of misdiagnosis decreases. The benefits of coaching are similar to those for scaffolding: Coaching provides focused help at critical times and only as much help as needed. In the best cases, it can provide new ways to see what you are doing, which can help students out of ruts. Ideally, coaching, like scaffolding, should fade as students become more expert. But both computer and human coaching is expensive to provide, and so it must have high payoffs to be worth the cost.

Articulation

Teachers have a variety of methods for getting students to articulate their ideas and thinking processes. For example, Bereiter and Scardamalia (1987) have students describe their thinking processes while planning an essay. Schoenfeld (1985) has students work in groups to solve difficult math problems so they are forced to articulate their thinking to each other. Inquiry teachers (Collins & Stevens, 1982, 1983) pose problems and questions for students to get them to articulate and refine their theories. As Brown (1985) pointed out, programs like Robot Odyssey force students to articulate their theories to construct robot agents to carry out their plans. These kinds of articulation help students formulate their ideas in a way that makes them available on other occasions.

The cost of articulation is that students may learn to talk a good game without really understanding. Also, emphasis on articulation discriminates against the less articulate, who might be able to do tasks perfectly well without any articulation. One benefit of articulation is that it helps make people's tacit knowledge explicit, and hence it is more available. Another benefit is that articulation allows people to see how other people think about the same problem. Making knowledge more available through articulation fosters transfer of that knowledge to new situations.

Reflection

Reflection involves looking back over one's performance on a task and comparing it to other people's performances, both good and bad, on similar tasks. This exploits the method of perceptual learning (Bransford, Franks, Vye, & Sherwood, 1989). For example, one can use reflective tape to mark critical parts of an athlete's body and videotape his or her performance in swinging a racket or throwing a javelin. Then it is possible to compare how the athlete's body moves during more and less successful performances, and how he or she moves compared with other athletes. Collins and Brown (1988) called this an *abstracted replay*; it allows the student to reflect systematically on the process. Another form of reflection is possible in Algebraland (J. Brown, 1985) or the Geometry Tutor (Anderson et al., 1985), where the system keeps a record of all the student moves in solving an algebra equation or developing a geometry proof. These reifications of the problem-solving process allow similar kinds of reflection.

The costs of reflection are that students often find it tedious to have to look back at their performance, and usually do not have the patience to try to improve their performance. Most students just want to do an activity and then move on to other activities. The benefits of reflection are that students have a chance to see processes for the first time, much like their first exposure to a mirror, and to compare their ways of doing things to other people's ways. Because they can see themselves from a new angle, students begin to develop new ways of seeing and talking about what they do. I particularly recommend the kinds of abstracted replays and reifications described earlier (Collins & Brown, 1988), and the cycle of performing, reflecting, and reperforming embodied in Arts Propel (Gardner, 1991; Wolf, 1987).

CONCLUSION

These are my candidate set of issues that designers should be concerned about, omitting issues about the knowledge learned (Collins et al., 1989) and the social settings in which learning occurs (Collins, Greeno, & Resnick, 1994), which I address elsewhere. By taking a cost–benefit approach to these issues, there is a chance that designers will be able to minimize the costs and maximize the benefits of any design decisions.

ACKNOWLEDGMENTS

This work was supported by the Center for Technology in Education under Grant No. 1–135562167-A1 from the Office of Educational Research and Improvement, U.S. Department of Education to Bank Street College of Education. I thank Roy Pea for his suggestion of adding Access vs. Understanding and Depth vs. Breadth of Knowledge to the set of issues.

REFERENCES

Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science, 228*, 456–468.

Benezet, L. P. (1991, May). The teaching of arithmetic: The story of an experiment. *Humanistic Mathematics Newsletter, 6*, 2–14.

Bereiter, C., & Scardamalia, M. (1987). *The psychology of written composition*. Hillsdale NJ: Lawrence Erlbaum Associates.

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). New York: Cambridge University Press.

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions. *Journal of the Learning Sciences, 2*(2), 141–178.

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

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

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

Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 79–98). New York: Academic Press.

Carver, S. M. (1990, April). *Integrating interactive technologies into classrooms: The Discover Rochester project*. Paper presented at the annual meeting of the American Educational Research Association, Boston.

Collins, A. (1990). Reformulating testing to measure learning and thinking. In N. Frederiksen, R. Glaser, A. Lesgold, & M. Shafto (Eds.), *Diagnostic monitoring of skills and knowledge acquisition* (pp. 325–350). Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A. (1991). Cognitive apprenticeship and instructional technology. In L. Idol & B. F. Jones (Eds.), *Educational values and cognitive instruction: Implications for reform* (pp. 119–136). Hillsdale, NJ: Lawrence Erlbaum Associates.
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.
Collins, A., Greeno, J. G., & Resnick, L. B. (1994). Learning environments. In T. Husen & T. N. Postlethwaite (Eds.), *International encyclopedia of education* (2nd ed., pp. 3297–3302). Oxford, UK: Pergamon.
Collins, A., Hawkins, J., & Carver, S. M. (1991). A cognitive apprenticeship for disadvantaged students. In B. Means, C. Chelemer & M. S. Knapp (Eds.), *Teaching advanced skills to at-risk students* (pp. 216–243). San Francisco: Jossey-Bass.
Collins, A., & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.), *Advances in instructional psychology* (Vol. 2, pp. 65–119). Hillsdale, NJ: Lawrence Erlbaum Associates.
Collins, A., & Stevens, A. L. (1983). A cognitive theory of interactive teaching. In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview* (pp. 247–278). Hillsdale, NJ: Lawrence Erlbaum Associates.
Cuban, L. (1984). *How teachers taught*. New York: Longman.
Frederiksen, J. R. & White, B. Y. (1990). Intelligent tutors as intelligent testers. In N. Frederiksen, R. Glaser, A. Lesgold, & M. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition* (pp. 1–25). Hillsdale, NJ: Lawrence Erlbaum Associates.
Gardner, H. (1991). Assessment in context: The alternative to standardized testing. In B. Gifford & C. O'Connor (Eds.), *Future assessments: Changing views of aptitude, achievement, and instruction* (pp. 77–120). Boston: Kluwer.
Harel, I. (1990). Children as software designers: A constructionist approach for learning mathematics. *The Journal of Mathematical Behavior, 9*(1), 3–93.
Heath, S. B. (1991). "It's about winning!" The language of knowledge in baseball. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 101–126). Washington, DC: American Psychological Association.
Hirsch, E. D., Jr. (1987). *Cultural literacy: What every American needs to know.* Boston: Houghton Mifflin.
Lampert, M. (1986). Knowing, doing, and teaching multiplication. *Cognition and Instruction, 3*, 305–342.
Lepper, M. R., Aspinwall, L., Mumme, D., & Chabay, R. W. (1990). Self-perception and social perception processes in tutoring: Subtle social control strategies of expert tutors. In J. M. Olson & M. P. Zanna (Eds.), *Self-inference and social inference: The Ontario Symposium* (Vol. 6, pp. 217–237). Hillsdale, NJ: Lawrence Erlbaum Associates.
Lesgold, A., Lajoie, S., Logan, D., & Eggan, G. (1990). Applying cognitive task analysis and research methods to assessment. In N. Frederiksen, R. Glaser, A. Lesgold, & M. Shafto (Eds.), *Diagnostic monitoring of skills and knowledge acquisition* (pp. 325–350). Hillsdale, NJ: Lawrence Erlbaum Associates.
Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and monitoring activities. *Cognition and Instruction, 1*, 117–175.
Reigeluth, C. M. (Ed.). (1983). *Instructional-design theories and models: An overview of their current status*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Reigeluth, C. M. (Ed.). (1987). *Instructional theories in action: Lessons illustrating selected theories and models*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Resnick, L. B., Bill, V. L., Lesgold, S. B., & Leer, M. N. (1991). Thinking in arithmetic class. In B. Means, C. Chelemer, & M. S. Knapp (Eds.), *Teaching advanced skills to at-risk students* (pp. 27–53). San Francisco: Jossey-Bass.
Schoenfeld, A. J. (1985). *Mathematical problem solving*. New York: Academic Press.
Stevens, A., & Roberts, B. (1983). Quantitative and qualitative simulation in computer-based training. *Journal of Computer-Based Instruction, 10*, 16–19.
Stigler, J. W., & Perry, M. (1988). Mathematics learning in Japanese, Chinese, and American classrooms. In G. Saxe & M. Gearhart (Eds.), *Children's mathematics* (pp. 27–54). San Francisco: Jossey-Bass.
White, B. Y., & Frederiksen, J. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 24*, 99–157.
Wiggins, G. (1989, May). A true test: Toward more authentic and equitable assessment. *Phi Delta Kappan*, pp. 703–713.
Wolf, D. P. (1987, December). Opening up assessment. *Educational Leadership*, pp. 24–29.