22

MICROWORLDS

Lloyd P. Rieber
The University of Georgia

22.1 MICROWORLDS

The introduction and spread of computer technology in schools since about 1980 have led to a vast assortment of educational software. Most of this software is instructional in nature, based on the paradigm of "explain, practice, and test." However, another, much smaller collection of software, known as *microworlds,* is based on very different principles, those of invention, play, and discovery. Instead of seeking to give students knowledge passed down from one generation to the next as efficiently as possible, the aim is to give students the resources to build and refine their own knowledge in personal and meaningful ways. The epistemology underlying microworlds is known as constructivism (Jonassen, 1991b). Once considered a peripheral movement in education, constructivist approaches to learning and education are now more widely endorsed and increasingly viable, due largely to advances in computer technology. While not negating the role of instruction, constructivist perspectives place central importance on a person's interaction in a domain and the relationship of this interaction with the person's prior knowledge.[1] A constructivist learning environment is characterized by students learning through active engagement, with encouragement, support, and resources to enable them to construct and communicate what they know and how they know it to others in a social context (Tinker & Thornton, 1992).

Constructivist approaches are not new to education. The progressive education ideals of John Dewey (e.g., 1916) are but one example. One of the reasons for the success of constructivist influences in education today, and perhaps the lack of success by Dewey in the first half of the twentieth century, is the widespread availability of resources that lead to rich explorations within a domain. Until only recently, it was not possible to give all students the kinds of interactive experiences in complex domains such as mathematics, physics, and biology that permit them to explore and invent in ways similar to those of mathematicians, physicists, and biologists. The technology of paper and pencil is limited to textual explanations and static drawings, thus limiting the way in which a domain can be represented and experienced. Historically, differential equations were the principal tool scientists used to study dynamic models. Such limits in representation likewise limit access to a domain's most advanced ideas to those few fortunate individuals who either have learning or metacognitive styles that are aligned with those representations or enjoy a socioeconomic status with resources and attitudes that offset such limitations to learning (Eccles & Wigfield, 1995). But the technology of computers affords a wider array of representations and experiences as well as greater availability to more people, beginning with even very young children (Resnick, 1999).

The purpose of this chapter is to review the theory and research of microworlds. The microworld literature can be confusing at times, making it difficult to distinguish microworlds from other forms of interactive software. Indeed, the term *microworld* is not used consistently even by members within the constructivist community itself. Other terms often used are *computational media* (diSessa, 1989), *interactive simulations* (White, 1992), *participatory simulations* (Wilensky & Stroup, 2002), and *computer-based manipulatives* (Horwitz & Christie, 2002). Therefore, different interpretations are reviewed, with the goal of teasing out essential characteristics of microworlds—theoretical and physical—and their relationship to other computer environments with which they are frequently compared and confused, such as computer-based simulations. Many issues remain contentious among those in the microworld community,

[1]Many people who ascribe to these learning principles do not necessarily characterize themselves as constructivists. See other chapters in this book for examples. Regardless, microworlds are rightly placed within a constructivist framework, if only for historical reasons.

such as model using versus model building (Feurzeig & Roberts, 1999; Penner, 2000/2001) and encouraging the use of computational media (i.e., those that require programming structures) versus tools with icon-based, or "point and click," interfaces (diSessa, Hoyles, Noss, & Edwards, 1995a).

Yet there is strong consensus on several key points within virtually all of the microworld literature. Computer-based microworlds offer the means to allow a much greater number of people, starting at a much younger age, to understand highly significant and applicable concepts and principles underlying all complex systems (e.g., White & Frederiksen, 1998). Two scientific principles deserve special mention: the vast array of rate of change problems common to all dynamic systems (Ogborn, 1999; Roschelle, Kaput, & Stroup, 2000) and decentralized systems, such as economics, ecosystems, ant colonies, and traffic jams (to name just a few), which operate on the basis of local objects or elements following relatively simple rules as they interact, rather than being based on a centralized leader or plan (Resnick, 1991, 1999). Qualitative understanding based on building and using concrete models is valued and encouraged. Indeed, many feel that the distinction between the classic concrete and the formal operations of Piaget's developmental learning theory becomes blurred and less important when students are given ready access and guidance in the use of computer-based microworlds (Ogborn, 1999). Finally, there is a reduction in the distance among learning science, doing science, and thinking like a scientist. Learning based on scientific inquiry is championed throughout the literature (again, for an example, see White & Frederiksen, 1998).

An historical context is used in this review due to the way in which advances in computer technology have directly influenced the development of microworlds. This review begins with work reported around 1980 and proceeds up to the present. The year 1980 is chosen for two reasons. First, it marks a profound juncture of education and technology—the approximate arrival and spread of the personal computer in homes and the classroom. This was the time at which the Apple computer company had begun aggressively marketing personal computers to education. The Apple II had just been introduced. The time was marked by a fascination with and enthusiasm about the potential of technology in education. Although serious work in educational computing had begun in the 1960s, the advent of the personal computer around 1980 made it possible for the first time for public-school educators to use a computer in the average classroom.

Second, the year 1980 marked the publication of a controversial book by Seymour Papert—*Mindstorms*—that offered a very different vision of education afforded by the burgeoning technology. In contrast to the emphasis on computer-assisted instruction that had dominated computer-based education up to that time (e.g., Suppes, 1980), Papert's vision focused on turning the power of the computer over to students, even those in elementary school, through computer programming. Although many computer languages were commonly used in schools around 1980, such as Pascal and BASIC, Papert and a team of talented individuals out of the Massachusetts Institute of Technology and Bolt, Baranek, and Newman began developing a radically different programming language in 1968, with support from the National Science Foundation, based on a procedural language called Lisp (short for list processing) (Feurzeig et al., 1969; cited in Abelson, 1982). They called their new language Logo, derived from the Greek word meaning "thought" or "idea." Logo was distinguishable from other languages by how its design was influenced by a particular philosophy of education:

> Logo is the name for a philosophy of education and for a continually evolving family of computer languages that aid its realization. Its learning environments articulate the principle that giving people personal control over powerful computational resources can enable them to establish intimate contact with profound ideas from science, from mathematics, and from the art of intellectual model building. Its computer languages are designed to transform computers into flexible tools to aid in learning, in playing, and in exploring. (Abelson, 1982, p. ix)

Logo was particularly distinguished from other programming languages by its use of turtle geometry. Users, as young as preschoolers, successfully learned to communicate with an object called a "turtle," commanding it to move around the screen or on the floor using commands such as FORWARD, BACK, LEFT, and RIGHT. As the turtle moved, it could leave a trail, thus combining the user's control of the computer with geometry and aesthetics. Logo was deliberately designed to map onto a child's own bodily movements in space. By encouraging children to "play turtle," thousands of children learned to control the turtle successfully in this way.

Of course, many other microworlds have become available since 1980. Besides Logo, this chapter reviews other examples in detail, including Boxer (diSessa, Abelson, & Ploger, 1991), ThinkerTools (White, 1993), SimCalc (Roschelle et al., 2000), and GenScope (Horwitz & Christie, 2000). However, because the goal of this chapter is to review research associated with microworlds in education, lengthy technical descriptions of these programs have been omitted. Other examples of microworlds not specifically examined in this chapter include Model-IT (Jackson, Stratford, Krajcik, & Soloway, 1996; Spitulnik, Krajcik, & Soloway, 1999), StarLogo (Resnick, 1991, 1999), Geometer's Sketchpad (Olive, 1998), Function Machine (Feurzeig, 1999), and Stella (Forrester, 1989; Richmond & Peterson, 1996). The work cited in this chapter represents just a fraction of the work that has been carried out in this area. Although microworld research and development is approaching 40 years of sustained effort (if you begin with Logo's emergence in the mid-1960s), it remains fresh and intriguing, advancing in step with the technology that supports it. Whether microworlds and the pedagogy that underlies them will eventually become a dominant approach in schools remains, unfortunately, a question left to speculation.

Research with microworlds has occurred during an interesting and somewhat tumultuous time in the history of educational research. Since 1980, educational research has broadened considerably to include a wide range of acceptable research methodologies. When researchers first took an interest in studying Logo, educational research was strongly dominated by quantitative, experimental research. In contrast, many of the early reports on Logo were anecdotal while, at the same time, written with enthusiasm about the technology's capabilities and potential, leading to hypish claims for their power and utility.

For example, Logo advocates suggested that it would "revolutionize" education, claims that now have the benefit of 20 years of scrutiny. These early promises, associated with data lacking scientific rigor, led to unfortunate battle lines being drawn between proponents and opponents of using microworlds and other constructivist approaches in education (an example is Tetenbaum & Mulkeen, 1984). Contemporary educational research has slowly shifted to accept alternative methods, mostly qualitative, led in part by technology-driven interpretations of the science of learning. Microworld research particularly is characterized by a history of multiple methods, of which the "design experiment" is the newest to be recognized (Barab & Kirshner, 2001; Brown, 1992; Collins, 1992; Edelson, 2002). The recent rise and formalization of design experiments are discussed later in this chapter.

22.1.1 Historical Origins of the Microworld Concept

The formal conception of a microworld, at least that afforded by computer technology, can be traced at least as far back as a chapter by Seymour Papert (1980a) in a seminal book edited by Robert Taylor entitled *The Computer in the School: Tutor, Tool, Tutee*. Papert's contribution was to the "tutee" section, that of the "computer as learner," or computer programming.[2] Papert (1980a) first defined a microworld as a

> . . . subset of reality or a constructed reality whose structure matches that of a given cognitive mechanism so as to provide an environment where the latter can operate effectively. The concept leads to the project of inventing microworlds so structured as to allow a human learner to exercise particular powerful ideas or intellectual skills. (p. 204)

Papert clearly tried to establish the idea that a microworld is based to a large degree on the way in which an individual is able to use a technological tool for the kinds of thinking and cognitive exploration that would not be possible without the technology. In his chapter, Papert also made it clear that the concept of a microworld was not new and related the idea to the longstanding use of math manipulatives, such as Cuisenaire rods. But Papert predicted that the availability of microcomputation offered the potential for radically different learning environments to be created and adopted throughout schools. Given the benefit of more than 20 years of educational hindsight, it is tempting to be amused at Papert's naiveté. After all, the history of educational technology is filled with examples of new technologies promising similar opportunities to transform education (Saettler, 1990). Yet Papert's focus on the individual learner as contributing to the definition of a microworld distinguishes his idealism from most of the other educational innovations that had already come and gone (Cuban, 1986, 2001).

The publication of *Mindstorms* in 1980 had a large impact on educational thinking and even a modest influence on educational practice—Logo classes for teachers filled to capacity in colleges of education across the country. This was due, again, partly to the confluence of education and technology at that time in history—there was little else available in the just-emerging educational computing curriculum. But *Mindstorms* laid out a compelling and provocative account of how computers might be used as part of the learning enterprise. It harshly criticized everything traditional in education and computing. Papert (1980b) took issue with most forms of formal instruction and imagined the computer providing a source of learning experiences that would allow a child to learn in ways that were natural and not forced:

> It is not true to say that the image of a child's relationship with a computer I shall develop here goes far beyond what is common in today's schools. My image does not go beyond: It goes in the opposite direction. (p. 5)

On one hand, Papert's criticism might have helped polarize discussions about the role of technology in education, leading to factions for and against Logo, and hence for and against constructivist approaches to learning, in the schools. It could even be argued that such polarizations slowed the adoption of technology in general in schools. On the other hand, Papert's insistence that the learning environments represented by Logo offered something entirely new helped clarify differences between merely assimilating the affordances of computers into the conventional curricula and teaching approaches and changing how education happens given technology.

Despite the apparent radicalism in these early writings, Papert, unlike others writing about Logo, was not fanatical, only provocative. Though naive about education, he was not naive about learning and a learner's need for support structures. For example, he makes one other interesting point in his chapter in Taylor's book, that of how a microworld must contain design boundaries:

> The use of the microworlds provides a model of a learning theory in which active learning consists of exploration by the learner of a microworld sufficiently bounded and transparent for constructive exploration and yet sufficiently rich for significant discovery. (Papert, 1980a, p. 208)

This is a telling statement because it foreshadows much of the later controversy over the role and nature of the boundaries of microworld design and whether *instructional* design could assume any place in it. While it demonstrates the importance Papert placed on exploration and discovery learning, it also shows his early acceptance of the need for a teacher or a microworld designer to identify boundaries for learning, thus contradicting the many criticisms made over the years thereafter that Papert thought that education and learning should be a "free for all" without guidance or interventions. Papert may be guilty of underestimating the difficulty of designing such boundaries, especially identifying where the boundaries lie for a particular child in a particular domain, but he certainly recognized

[2]Papert (1980b) later included a revised and longer version of this chapter in the provocative book *Mindstorms*. Although it is in *Mindstorms* that Papert more forcefully argued for a microworld to be a legitimate alternative learning environment to that of traditional classroom practice, I find Papert's writing in Taylor's book to be much clearer and more direct.

the need for guidance, both in the microworld itself and in the teacher's assistance to a child using it. As Papert (1980a) writes,

> The construction of a network of microworlds provides a vision of education planning that is in important respects "opposite" to the concept of "curriculum." This does not mean that no teaching is necessary or that there are no "behavioral objectives." But the relationship of the teacher to learner is very different: the teacher introduces the learner to the microworld in which discoveries will be made, rather than to the discovery itself. (p. 209)

In his book *The Children's Machine,* published over a decade later, in 1993, Papert continued to explore the issue of the use and misuse of the "curriculum" and the teacher's pivotal role in the learning enterprise. Papert admitted to having little contact with teachers before *Mindstorms* and believed that teachers would be among the most difficult obstacles in transforming education given the technology. He expected very few teachers to read it. However, at the time hundreds of thousands of teachers *were* reading it, giving him a "passport into the world of teachers" (Papert, 1993), and helped change his earlier conceptions:

> . . . My identification of "teacher" with "School" slowly dissolved into a perception of a far more complex relationship. The shift brought both a liberating sense that the balance of forces was more favorable to change than I had supposed and, at the same time, a new challenge to understand the interplay of currents in the world of teachers that favor change and that resist it. Finding ways to support the evolution of these currents may be among the most important contributions one can make to promote educational change. (p. 59)

According to Papert (1980b), the proper use of the computer for learning was in the child's total appropriation of it via learning to program:

> Once programming is seen in the proper perspective, there is nothing very surprising about the fact that this should happen. Programming a computer means nothing more than communicating to it in a language that it and the human user can both "understand." And learning languages is one of the things children do best. Every normal child learns to talk. Why then should a child not learn to "talk" to a computer? (pp. 5–6)

For Papert, the difficulties in learning to program a computer stemmed not from the difficulty of the task, but from the lack of context of learning to do so, especially in the programming means available to the child. Not surprisingly, Papert, educated as a mathematician, was interested in finding ways for children to learn mathematics as naturally as they acquired language early in life. Similar to the idea that the best way to learn Spanish is to go and live in Spain, Papert conjectured that learning mathematics via Logo was similar to having students visit a computerized Mathland where the inhabitants (i.e., the turtle) speak only Logo. And because mathematics is the language of Logo, children would learn mathematics naturally by using it to communicate to the turtle. In Mathland, people do not just study mathematics, according to Papert, they "live" mathematics.

Papert's (1980a) emphasis on the learner's interaction with a microworld was rooted in Piagetian learning theory:[3]

> The design of microworlds reflects a position in genetic epistemology: in particular a structuralist and constructivist position derived from Piaget that attaches great importance to the influence on the developed forms of the developmental path. (p. 208)

Interestingly, of the two principal parts of Piaget's developmental learning theory, Papert focused a great deal on one and almost ignored the second (Clements, 1989). He emphasized the stage-independent part of Piaget's theory, based on the process of equilibration, and the enabling mechanisms of assimilation and accommodation. In contrast, little attention was given to the stage-dependent part of Piaget's theory, suggesting that all people follow an invariant progression of intellectual development from birth, starting with the sensorimotor and ending with formal operations. Indeed, Papert and his colleagues felt that too much of formal education valued the formal and abstract, and too little valued the concrete.

Experience with any of the microworlds described in this chapter will lead one to see that all microworlds directly support acquiring a qualitative understanding of a problem in terms that are developmentally appropriate for a child, yet also are clearly connected to the formal, rigorous mathematics side of the domain. This value placed on the concrete and qualitative aspects of understanding permeates all of the microworld literature to the present day (see Papert's [1993, p. 148] criticism of the "supervaluation of the abstract"). This is consistent with longstanding research that indicates that novices and experts often use a qualitative approach to solve problems (Chi, Feltovich, & Glaser, 1981). Papert did not undervalue the formal and abstract side of a domain but, rather, tried to bring up to at least an equal standing the importance of an individual being able to connect to the domain through concrete, qualitative means.

Using language from Piaget's work, Papert referred to the use of the turtle as a "transitional" object, connecting what the child already knows to the domain of geometry. This is made possible by the fact that the child, like the turtle, has two attributes in common—a position and a heading. For example, a child can "play turtle" to figure out how to make the turtle draw a circle by first walking in a circle and describing the activity. The child soon learns that a circle is made by repeating a pattern of moving forward a little, followed by turning a little. Thinking of a curve in this fashion is a fundamental concept of differential calculus. Transitional objects become more sophisticated over time. A professional mathematician will construct diagrams for exactly the same purpose (which, for Papert, are also examples of microworlds). But, in all cases, such use of microworlds can be viewed as "genetic stepping stones" (Papert, 1980b, p. 206) from the learner's current understanding (without the microworld) to the internalization of powerful ideas (differential calculus) with the help of the microworld.

Mindstorms contained several fundamental ideas that continue to thrive in the vocabulary and thinking of current constructivist conceptions of learning. Among the most profound is the idea of an *object to think with,* the Logo turtle, of course, being a prime example. Thus, the turtle becomes a way for the child to grapple with mathematical ideas usually considered

[3]Papert spent 5 years studying with Piaget in Geneva, Switzerland.

too difficult or abstract. A prime role served by the turtle is the way it "concretizes" abstract ideas. A classic example is when a child learns that the number "360" has special properties in geometric space. Making a square by repeating four times the commands FORWARD 50 RIGHT 90 shows a concrete relationship between the square and 360. This idea can be expanded so that all other regular polygons can be constructed by dividing 360 by the number of sides desired.

Another important microworld idea is that of *debugging*. While obviously rooted in the process of computer programming, debugging is really concerned about learning from one's mistakes. Unlike conventional education, where errors are to be avoided at all costs, errors in problem-solving tasks such as programming are unavoidable and therefore expected. Errors actually become a rich source of information, without which a correct solution could not be found. The use of an external artifact, such as a computational microworld, as an object to think with to extend our intellectual capabilities, coupled with a learning strategy of expecting and using errors made as a route to successful problem solving, is an integral part of all contemporary learning theories (Norman, 1993; Salomon, Perkins, & Globerson, 1991).

So, as we have seen in this brief historical overview, the concept of a microworld became firmly established as a place for people of all ages to explore in personally satisfying ways complex ideas from domains usually considered intellectually inaccessible to them. These same ideas continue to be championed today, as the following contemporary definition of a microworld by Andy diSessa (2000), one of constructivism's most vocal and articulate advocates since Papert, shows:

> A microworld is a genre of computational document aimed at embedding important ideas in a form that students can readily explore. The best microworlds have an easy-to-understand set of operations that students can use to engage tasks of value to them, and in doing so, they come to understanding powerful underlying principles. You might come to understand ecology, for example, by building your own little creatures that compete with and are dependent on each other. (p. 47)

Of all the possible definitions of a microworld, perhaps the most elegant comes from Clements (1989): "A microworld is a small playground of the mind" (p. 86). In the next section, we consider characteristics of microworlds that provide playful opportunities for learning.

22.2 GENERAL CHARACTERISTICS OF MICROWORLDS

So, what makes a microworld a microworld? Is it a collection of software components or characteristics, or something more? Microworlds are part of a larger set or approach to education known as *exploratory learning* (diSessa, Hoyles, Noss, & Edwards, 1995a). All exploratory learning approaches are based on the following four principles: (a) Learners can and should take control of their own learning; (b) knowledge is rich and multidimensional; (c) learners approach the learning task in very diverse ways; and (d) it is possible for learning to feel natural and uncoaxed, that is, it does not have to be forced or contrived. These are idealistic pursuits, to say the least. These principles lead to some interesting educational outcomes or issues. For example, there is no "best approach" to teach something (at least for all but the most narrow of skill sets), nor is there a "best way" to learn. The goals of education should focus on complex learning outcomes, such as problem solving, where depth of understanding, not breadth of coverage, is valued. Furthermore, student learning should be based, at least partially, on student interests. This implies that adequate time and resources must be given to students to pursue ideas sufficiently before they are asked to move on to other educational goals. Another outcome is also very much implied: Support and resources for learning are equally diverse, coming in forms such as other people and the full range of technological innovations, including the computer, of course, but also paper and pencil. This, in turn, suggests a very social context for learning and it is expected that the personal interests of students will be tied to social situations.

There are many examples of interactive, exploratory learning environments in education. Examples include the range of hypertext and hypermedia (Jonassen, 1991a, 1992) (including the World Wide Web) and interactive multimedia (such as simulations and games). However, microworlds can be distinguished from other kinds of exploratory learning environments by their focus on immersive learning and their sensitive tuning to a person's cognitive and motivational states. It is debatable whether a software program can be rightly called a microworld based solely on the software's physical and design attributes. However, a structural view attempts to do just that by identifying a list of features, characteristics, or design attributes common to the category of software commonly labeled a microworld. Thus, if other software shares these features, one could rightly define it as a microworld. A microworld, using such a structural definition, would, according to Edwards (1995), consist of the following.

- A set of computational objects that model the mathematical or physical properties of the microworld's domain
- Links to multiple representations of the underlying properties of the model
- The ability to combine objects or operations in complex ways, similar to the idea of combining words and sentences in a language
- A set of activities or challenges that are inherent or preprogrammed in the microworld; the student is challenged to solve problems, reach a goal, etc.

While such *structural* affordances are important, the true tests of a microworld are *functional*—whether it provides a legitimate and appropriate doorway to a domain for a person in a way that captures the person's interest and curiosity (Edwards, 1995). In other words, for an interactive learning environment to be considered a microworld, a person must "get it" almost immediately—understand a simple aspect of the domain very quickly with the microworld—and then *want* to explore the domain further with the microworld (Rieber, 1996). Again, the analogy of choice for Papert was language learning because learning most math and science offers the same richness and complexity as learning a foreign language.

A functional view is based on the dynamic relationship among the software, the student, and the setting. Whether or not the software can be considered a microworld depends on this interrelationship when the software is actually used. Students are expected to be able to manipulate the objects and features of the microworld "with the purpose of inducing or discovering their properties and the functioning of the system as a whole" (Edwards, 1995, p. 144). Students are also expected to be able to interpret the feedback generated by the software based on their actions and modify the microworld to achieve their goal (i.e., debugging). And students are expected to "use the objects and operations in the microworld either to create new entities or to solve specific problems or challenges (or both)" (Edwards, 1995, p. 144).

Therefore, a microworld must be defined at the interface between an individual user in a social context and a software tool possessing the following five functional attributes:

- It is domain specific;
- it provides a doorway to the domain for the user by offering a simple example of the domain that is immediately understandable by the user;
- it leads to activity that can be intrinsically motivating to the user—the user wants to participate and persist at the task for some time;
- it leads to immersive activity best characterized by words such as play, inquiry, and invention; and
- it is situated in a constructivist philosophy of learning.

The fifth and final attribute demands that successful learning with a microworld assumes a conducive classroom environment with a very able teacher serving a dual role: teacher-as-facilitator and teacher-as-learner. The teacher's role is critical in supporting and challenging student learning while at the same time modeling the learning process with the microworld. It is important to note, perhaps surprisingly, that the principles of microworlds discussed in this section do not require that they be computer based. A child's sandbox with a collection of different-sized buckets can be considered a microworld for understanding volume. In mathematics, the use of manipulatives, such as Cuisenaire rods, can be a microworld for developing an understanding of number theory. But computational media provide unprecedented exploratory and experiential opportunities.

In summary, while both structures and functions of a microworld are important, a functional orientation is closer to the constructivist ideals of understanding interactions with technology from the learner's point of view. Of course, this means that the same software program may be a microworld for one person and not another. Microworlds can be classified as a type of cognitive tool in that they extend our limited cognitive abilities, similar to the way in which a physical tool, like a hammer or saw, extends our limited physical abilities (Jonassen, 1996; Salomon et al., 1991). However, microworlds are domain specific and carry curricular assumptions and pedagogical recommendations for how the domain, such as mathematics or physics, ought to be taught.

22.3 MICROWORLD RESEARCH WITH LOGO

To understand early research efforts involving Logo, one must understand the educational research climate at the time. Educational research around 1980 was dominated by experimental design. This, compounded with the long-standing view that media directly "affects" learning (for a summary see Clark, 1994, 2001; Kozma, 1994), led Papert to challenge the research questions being asked at the time and what methodologies were being used to generate, analyze, and interpret the data. Not surprisingly, Papert (1987) was critical of the controlled experiment in which everything except one variable is controlled and studied: "I shall argue that this is radically incompatible with the enterprise of rebuilding an education in which nothing shall be the same" (p. 22). He complained that criticism against the computer was "technocentric" in that it focused on the technology, not the student. Such a view likens computers and Logo to agents that act directly on thinking and learning and is characterized by research questions about the "effects" of computers or Logo on learning:

> Consider for a moment some questions that are "obviously" absurd. Does wood produce good houses? If I built a house out of wood and it fell down, would this show that wood does not produce good houses? Do hammers and saws produce good furniture? These betray themselves as technocentric questions by ignoring people and the elements only people can introduce: skill, design, aesthetics. (Papert, 1987, p. 24)

Papert contended that these were similar to the kinds of questions being asked about the computer and Logo at the time (circa 1986). Logo, Papert (1987) said, was not like a drug being tested in a medical experiment but, instead, needed to be viewed as a cultural element: ". . . something that can be powerful when it is integrated into a culture but is simply isolated technical knowledge when it is not" (p. 24).

Papert (1987) sought to portray Logo as a "cultural building material" (p. 24). As an example, he presented the work of a teacher who had children "mess about with clocks" with the goal of trying to develop good ways to measure time. This teacher's science room was equipped with lots of everyday objects and materials—as well as computers. So the computer was just one more set of materials available to the students in their inquiry. For Papert, the way this teacher used Logo based on the students' own interests was in stark contrast to the kinds of uses of Logo that educational researchers were expecting to be studied. Papert (1987) believed that the computer must be viewed as part of the context or culture for human development: ". . . If we are interested in eliminating technocentrism from thinking about computers in education, we may find ourselves having to re-examine assumptions about education that were made long before the advent of computers" (p. 23).

Mainstream Logo research in the early 1980s was characterized by questions looking for "effects of Logo" on children's learning.[4] Probably the most careful and scholarly examples of this type of research were carried out by Douglas Clements, a mathematics educator at Kent State University. Clements conducted a series of Logo studies that investigated the effects of Logo programming on children's cognition, metacognition, and mathematical ability (examples include Clements [1984, 1986, 1987] and Clements & Gullo [1984]). He found that children working with Logo did, in fact, think differently about mathematics in deep and interesting ways. However, the results of research on whether this thinking transferred to non-Logo tasks were quite mixed. Again, the role of the teacher was central. For such transfer to occur, the teacher needed to create explicit links between the Logo activities and other mathematical activities. Clements (1987) showed that it was possible for master teachers to help students form broad mathematical understanding from their Logo activities.

In one particular study, often cited by early Logo enthusiasts, Clements studied the effects of learning Logo programming on children's cognitive style, metacognitive ability, cognitive development, and ability to describe directions. The goal was to look broadly for the types of influences that Logo programming was having on young children. He compared nine children who programmed with Logo for 12 weeks (two 40-min sessions per week) to another group of nine children who interacted with a variety of computer-assisted instruction (CAI) software packages. The rationale of such a comparison was that "... any benefits derived from computer programming can be attributed to interactive experiences with computers, rather than to the programming activity per se" (Clements & Gullo, 1984, p. 1052). It is easy to be confused today about what such a comparison would uncover, but it needs to be understood in the context of how new all of this technology was at the time. The study found very positive results favoring the Logo programming group. They outscored their CAI counterparts on virtually all measures (except cognitive development). Despite obvious methodological problems, such as the very limited sample size, Clements and Gullo concluded that the study provided evidence that programming may affect problem-solving ability and cognitive style.

Despite this positive outcome favoring Logo, Papert (1987) still felt that all such research missed the point as he critiqued the Clements and Gullo study and compared it to another done at Bank Street College (i.e., Pea & Kurland, 1984) that found negative results: "Both studies are flawed, though to very different extents, by inadequate recognition of the fact that what they are looking at, and therefore making discoveries about, is not programming but cultures that happen to have in common the presence of a computer and the Logo language" (p. 27). The work by Clements and his colleagues was carefully done and well thought out, yet clearly at odds with the philosophical intent of Logo.[5]

Some of the most interesting microworld research also began in the early 1980s, that done by Barbara White and her colleagues. What is most noteworthy about White's work is its consistent themes, which continue to the present day. Her early research, done in collaboration with Andy diSessa, focused on middle-school students learning physics with the "dynaturtle." The dynaturtle was an extension of the familiar Logo turtle, except that in addition to position and heading, it had the attribute of velocity—it was a "dynamic" turtle. That work led to White's (1984) dissertation research, in which she developed a series of game-like physics activities for students to explore, using Logo as an authoring tool to create these activities. In the early 1990s, she was instrumental in developing ThinkerTools, a physics modeling program suitable for elementary- and middle-school students. Accompanying the tool itself was a well-crafted pedagogical approach based on scientific inquiry. The ThinkerTools software and curriculum have continually evolved. ThinkerTools began by emphasizing how computer microworlds can facilitate learning physics and has evolved to emphasize helping students "to learn about the nature of scientific models and the process of scientific inquiry" (White & Frederiksen, 2000a, p. 321). Taken as a whole, it represents a thoughtful design and research effort. Another important aspect of White's work is the strong research program that has accompanied it. Her research results are widely cited by advocates of constructivist uses of computers.

Using the dynaturtle microworld, White conducted a series of investigations using a continually refined set of games that were designed to represent Newtonian motion phenomena clearly without unnecessary and distractive elements. Another goal was to help children focus on their own physics understanding in a reflective manner. The games she designed helped children to understand physics principles about which other research showed that they held firm misconceptions, such as the idea that objects eventually "run out of force." Interestingly, her research used a strong quantitative research methodology, comparing pretest and posttest scores of high-school students who used the computer games to those of a control group that did not. The results were very positive in favor of the dynaturtle games: Students who played the games improved their understanding of force and motion more than those who did not (White, 1984). Another interesting outcome of this line of research was the way it broadened the conception of a microworld from computer programming to interactions with "interactive simulations"[6] and modeling tools, of which ThinkerTools can be included as an example. We continue the discussion of Barbara White's work when we focus on ThinkerTools later in this chapter.

[4]For an additional review of early Logo research, see the chapter by Jonassen and Reeves in this volume.

[5]I had the same mindset as Clements at the time. I did a research project for my master's degree in 1983 that studied the "effects" of Logo (Rieber, 1987). It was a small study with limited exposure, yet I received over 300 requests for reprints, the most for any study I ever conducted. Such was the interest by the educational community in knowing more about what Logo was "doing to" our children.

[6]This particular study influenced my work to a great extent and led to my own research in the area of simulations and games (see Rieber, 1990, 1991; Rieber & Parmley, 1995).

22.3.1 The Emergence of a New Research Methodology: Design Experiments

The criticisms of educational research methodologies by Papert and many others in the Logo community led them to conduct field tests in cooperating schools. The formulation of partnerships between universities and schools with the desire to test a technological innovation without being restricted to the "rules" of prevalent research methods and curriculum constraints (i.e., not enough time and not enough resources) has become the preferred methodology of almost all of the microworld researchers and developers discussed in this chapter. The goal of all of these field tests is simultaneously to understand how the innovation works outside the team's rarefied development laboratories while also improving the innovation's design. This combination of a formative evaluation of the innovation (again, to improve it) and an analysis of the messy implementation process with real teachers and students has slowly led to a new research methodology called a *design experiment*. This research methodology, also referred to as design studies, design research, formative research, and development research (Richey & Nelson, 1996; van den Akker, 1999), differs from traditional educational research in which specific variables are rigidly controlled throughout an investigation. A design experiment sets a specific pedagogical goal at the beginning and then seeks to determine the necessary organization, strategies, and technological support necessary to reach the goal (Newman, 1990). Such experiments involve an iterative and self-correcting process that resolves problems as they occur. The process is documented to show what path was taken to achieve the goal, what problems were encountered, and how they were handled. Although the impact of an innovation on individual achievement is important, the unit of analysis in a design experiment is typically at the class or school level and includes social dynamics in the analysis. Vygotky's classic work on the zone of proximal development—what people can learn with and without aid—has been a clear influence on design experiments. Some of the first calls for design experiments in the early 1990s were based on the perceived need that technology would soon be adopted widely by schools, requiring a new methodology to cope with understanding what such implementation meant (Newman, 1990). Given the anticipated deluge, researchers needed to leave the laboratory and, instead, use schools themselves as their research venue.

In an early and seminal work, Collins (1992) described some of the problems and weaknesses of design experiments, at least as carried out up to that time. He cited the tendency for the researchers to be the designers of the innovation itself, hence being prone to bias due to their vested interest in seeing the innovation succeed. This also created the tendency to focus only on successful aspects of the innovation, with a temptation to exclude a wider examination of the innovation's use and implementation. The methodologies of design experiments varied widely, making it different to draw conclusions across the studies. Finally, design research is often carried out without a strong theoretical framework, thus making any results difficult to interpret. While the field has tried to solidify and elaborate on what a design experiment is and is not over the past decade, much remains to be done. It appears at present that design experiments are better viewed as explanatory frameworks for conducting research rather than clear methodologies.

In summary, the conceptual basis of design experiments and the methodology that is slowly emerging to accompany it appear to be aligned with the history and state of microworld research. Although the beginning articulation of design experiments is usually dated to the writings of Brown (1992) , Collins (1992), and Newman (1990, 1992), its "unarticulated" use predates these early works by at least 10 years, as it characterizes the abundance of the field research using Logo. Much of the other research on the microworlds described in the remaining sections of this chapter also resonates with design experiments, though this work has been poorly documented, consisting of internal memos and anecdotal reports within conceptual or theoretical publications. Fortunately, the methodology of design experiments is beginning to be recognized by the educational research community at large. This acceptance, especially among research journal editors, is likely to create a small revolution in the way in which research with innovative technology and students is conducted.

22.4 GOING BEYOND LOGO: BOXER

Boxer, according to diSessa et al. (1991), "is the name for a multipurpose computational medium intended to be used by people who are not computer specialists. Boxer incorporates a broad spectrum of functions—from hypertext processing, to dynamic and interactive graphics, to databases and programming—all within a uniform and easily learned framework" (p. 3). Boxer's principal designer and advocate is Andy diSessa, of the University of California at Berkeley. Boxer's roots are closely tied to those of Logo. Boxer originated while diSessa was at MIT and part of the Logo team. Despite diSessa's admiration of Logo and what it represented, he soon became dissatisfied with Logo's limitations (Resnick's motivation to create StarLogo was based on similar dissatisfactions with Logo's limitations). For example, Logo, though an easy language to start using, is difficult to master. Children quickly learn how to use turtle geometry commands to draw simple shapes, such as squares and triangles, and even complex shapes consisting of a long series of turtle commands, but it is difficult for most children to progress to advanced features of the language, such as writing procedures, combining procedures, and using variables. Another drawback of Logo is that it is essentially just a computer programming language, a variant of LISP, though with special features, such as turtle geometry. It is difficult for students and teachers to learn Logo well enough to program it to do other meaningful things, such as journal keeping and database applications. Finally, although Logo enjoyed much success with elementary- and middle-school students, it was difficult to "grow up" using Logo for advanced computational problems. Similarly, Logo was rarely viewed by teachers as a tool that they should use for their own personal learning or professional tasks. (See diSessa [1997] for other examples of how its design transcends that of Logo.)

diSessa sought to design a new tool to overcome these difficulties by creating not just another programming language, but a "computational medium." Again, Boxer and Logo share much in common as to educational philosophy and purpose. However, Boxer was designed to take advantage of all that had been learned from observing children using Logo up to the time the Boxer research group was formed in 1981. It was meant as a successor to Logo, not just a variant.

Boxer was designed based on two major principles related to learning: concreteness and the use of a spatial metaphor. Concreteness implies that all aspects and functions of the system should be visible and directly manipulable. The use of a consistent spatial metaphor capitalizes on a person's spatial abilities for relating objects or processes. For example, the principal object is a box, hence the name Boxer. A box can contain any element or data structure, such as text, graphics, programs, or even other boxes. The use of boxes allows a person to use intuitive spatial relations such as "outside," "inside," and "next" directly in the programming. Like Logo, Boxer has gone through a slow and serious development cycle of about 15 years, with much of this work best characterized as design experiments. It has been available on typical desktop computers for only a short period of time. Although it is difficult to predict technology adoption within education, Boxer has the potential for wide-scale use within K–12 schools, especially given its ability to adapt and extend to encompass data types and teaching and learning styles. Unfortunately, the question of whether Boxer will be adopted widely in education will probably be decided by factors other than those related to learning and cognition. Other, simpler multimedia authoring tools, such as HyperStudio and PowerPoint, have been marketed very successfully, due in part to their fit to more traditional uses of technology in education. Interestingly, the latest versions of Logo, such as Microworlds Pro, have incorporated many mainstream multimedia features to compete effectively in the education market.

Boxer makes it easy for teachers and students to build small-scale microworlds in many domains. An interesting example of how children can appropriate Boxer in unexpected ways is described by Adams and diSessa (1991). In this study, they showed how a classroom of children used a motion microworld given to them. The microworld required the student to input three pieces of data, corresponding to the turtle's initial position, speed, and acceleration. For example, if the students entered the numbers 0, 4, 0, the turtle started at the 0 position on a number line at an initial speed of 4 distance units per second. Since the acceleration is 0 (the third number), the turtle moved at this uniform speed forever. If the student entered 1, 3, 2, the turtle started moving with an initial speed of 3 distance units per second from the 1 position on the number. However, the speed increased by 2 distance units each second, thus the speed of the turtle generated a list of velocities (e.g., 3, 5, 7, 9, 11, etc.) and positions (1, 4, 9, 16, 25, 36, etc.) in 1-sec increments. In many ways, such a microworld can be considered as a simple physics model that could be written with almost any programming, authoring, or modeling software. However, a difference with Boxer is that all elements of the model remain changeable or manipulable at all times. As part of their research on how students would develop in their understanding of physics and Boxer, Adams and diSessa (1991) gave these students a problem that, unknown to them, was impossible to solve. The problem was to enter the triplets of data for each of two concurrently running turtles so that each would "pass" the other three times on the number line. There are no initial conditions that can be represented by these three numbers for each turtle that leads to such a motion. Transcripts of two students working on the problem showed their speculation that the problem could not be solved. But they soon wondered whether it was possible to alter the motion of the turtles directly by editing the velocity and position *lists* directly, thus bypassing the initial three data points. In a sense, such a direct method of manipulating the motion was cheating! However, Boxer allowed such a clever manipulation, thus also allowing the two students to reach a deeper understanding of motion. Adams and diSessa (1991) go on to describe how this technique was soon adopted by other students in the class, but through interesting negotiations with the teacher (i.e., it was permitted for difficult problems, but students were still expected to use the original method for simpler problems). Demonstrating the social dynamics of good ideas, Adams and diSessa (1991) explain: "This strategy spread in the classroom to become a communal resource for attacking the most difficult problems. The teacher and students negotiated ground rules for using these new resources productively. Although we did not plan this episode, we see it as an example of a kind of student-initiated learning that can emerge given a learning-oriented classroom and open technical designs" (pp. 88–89).

Boxer is interesting not only because of its own characteristics and affordances for learning, but also because of the history of its design within the microworld community. The roots of Boxer lie in criticisms and dissatisfactions with Logo, though diSessa and his colleagues are quick to respect all that Logo represents. Fortunately, they were willing to continue to "push the envelope" on the technology in ways that are consistent with the aims of Papert and other Logo pioneers. This is important because dissatisfaction with the state of microworld development is a powerful stimulus to improving it.

22.5 CONSTRUCTIONISM: MICROWORLD RESEARCH EVOLVES

Work with Logo in the constructivist community evolved beyond its philosophical roots in Piaget's constructivism to form a pedagogical approach called *constructionism,* a word coined by Papert (1991) to suggest another metaphor, that of "learning by building":

> Constructio$\underline{n}$ism—the N word as opposed to the V word—shares constructivism's connotation of learning as "building knowledge structures" irrespective of the circumstances of the learning. It then adds the idea that this happens especially felicitously in a context where the learner is engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe. (p. 1)

Constructionism is strongly rooted in student-generated projects. Projects offer a way critically to relate motivation and thinking and can be defined as "relatively long-term, problem-focused, and meaningful units of instruction that integrate concepts from a number of disciplines or fields of study"

(Blumenfeld et al., 1991, p. 370). Projects have two essential components: a driving question or problem and activities that result in one or more *artifacts* (Blumenfeld et al., 1991). Artifacts are "sharable and critiquable externalizations of students' cognitive work in classrooms" and "proceed through intermediate phases and are continuously subject to revision and improvement" (Blumenfeld et al., 1991, pp. 370–371).

It is important that the driving question not be overly constrained by the teacher. Instead, students need much room to create and use their own approaches to designing and developing the project. Projects, as external artifacts, are public representations of the students' solution. The artifacts, developed over time, reflect their understanding of the problem over time as well. In contrast, traditional school tasks, such as worksheets, have no driving question and, thus, no authentic purpose to motivate the student to draw or rally the difficult cognitive processes necessary for complex problem- solving.

A good example of an early constructionist research project was conducted by Harel and Papert (1990, 1991) as part of the Instructional Software Design Project (ISDP). This study is often cited among Logo and project-based learning proponents, so great attention to it is warranted here. The purpose of the ISDP was to give children the role of designer/producer of educational software rather than consumer of software. The research question of the study focused on ways in which children might use technology for their own purposes and how to facilitate children's reflection about what they are doing with technology. The study emphasized "developing new kinds of activities in which children can exercise their doing/learning/thinking" and "project activity which is self-directed by the student within a cultural/social context that offers support and help in particularly unobtrusive ways" (Harel & Papert, 1991, p. 42).

The study compared three classes: (a) 17 fourth-grade students who each worked with Logo for about 4 hr per week over a period of 15 weeks to design instructional software on fractions for use by another class (ISDP class); (b) 18 students who were also studying fractions and learning Logo, but not at the same time (control class 1); and (c) 16 students who were also studying fractions but not Logo (control class 2). Students were interviewed and tested on their understanding of fractions prior to the research. At the start of each work session, students in the ISDP group were required to spent 5–7 min writing their plans and drawing designs in their designer notebooks. The rest of the work session lasted approximately 50 min. Collaboration and sharing were encouraged. At the end of the session, students were required to write about problems and issues related to their projects confronted during the session. The projects were open-ended in the sense that students could choose whatever they wanted to design, teach, and program.

The study used both experimental and qualitative methodologies. All three classes were pretested on their knowledge of fractions and Logo. All students were then given a posttest at the end of the study. There were no significant differences among the three classes based on the pretest. During the study, observations (some videotaped) of and interviews with several students in the ISDP group were conducted, including an analysis of these students' designer notebooks and their finished projects. All 51 students were interviewed before and after the study. Students in the ISDP group outperformed the other two groups on the fractions test: ISDP, 74%; control class 1, 66%; and control class 2, 56%. Similarly, the ISDP group also outperformed the other students on questions from a standardized mathematics test related to fractions and rational numbers. (It is important to note that the ISDP group had additional, though not formal, exposure to fractions via several focus-group sessions.)

The qualitative results focused on four issues: (a) development of concept, (b) appropriation of project, (c) rhythm of work, and (d) cognitive awareness and control. The children's early development of the concept of fractions was very rigid and spatial. Their understanding was limited to very specific prototypes, such as half of a circle. By the end of the project their understanding was much more generalized and connected to everyday objects, especially outside of school. Many children resisted the task of designing software about fractions, but they all soon appropriated the task for themselves. The openness of what could constitute a design helped with this, as well as the encouragement of socialization as part of the design work. The fact that the children had access to computers to do their work on a daily basis was very important. It allowed them to migrate between periods of intense work and periods of playful, social behavior. Students in the experimental class became very metacognitively aware of their designs and work habits. They developed "problem-finding" skills. They became aware of strategies to solve problems and also learned to activate them. They developed the ability to discard bad designs and to search for better ones. They learned to control distractions and anxiety. They learned how to practice continual evaluation of designs in a social setting. They learned to monitor their solution processes and were able to articulate their design tasks.

Harel and Papert (1991) strongly suggest that what made a difference here was not Logo or any particular group of strategies but, rather, that a "total learning environment" (p. 70) was created that permitted a culture of design work to flourish. They particularly point to the affective influences of this environment. These students developed a different "relationship with fractions" (p. 71), that is, they came to like fractions and saw the relevancy of this mathematics to their everyday lives. Many reported "seeing fractions everywhere." Harel and Papert resist any tendency to report the success as being "caused" by Logo. Instead, "learning how to program and using Logo enabled these students to become more involved in thinking about fractions knowledge" (p. 73). They point to Logo's allowing such constructions about fractions to take place. The ISDP put students in contact with the "deep structure" of rational-number knowledge, compared to the surface structure that most school curricula emphasize.

Despite the positive outcome of this early constructionist research and the enthusiastic reporting by Harel and Papert, successful project-based learning is not a panacea. Success is based on many critical assumptions or characteristics and failure in any one can thwart the experience. Examples include an appreciation of the complex interrelationship between learning and motivation, an emphasis on student-driven questions or problems, and the commitment of the teacher and his/her willingness to organize the classroom to allow the complexities of project-based learning to occur and be supported (Blumenfeld et al., 1991). Fortunately, the recent and continuing development of rich technological tools directly support both teachers and students in the creation and sharing of artifacts.

Students must be sufficiently motivated over a long period to gain the benefits of project-based learning. Among the factors that contribute to this motivation are "whether students find the project to be interesting and valuable, whether they perceive that they have the competence to engage in and complete the project, and whether they focus on learning rather than on outcomes and grades" (Blumenfeld et al. 1991, p. 375). The teacher's role is critical in all this. Teachers need to create opportunities for project-based learning, support and guide student learning through scaffolding and modeling, encourage and help students manage learning and metacognitive processes, and help students assess their own learning and provide feedback. Whether teachers will be able to meet these demands depends in large part on their own understanding of the content embedded in projects, their ability to teach and recognition of student difficulty in learning the content (i.e., pedagogical awarenesses), and their willingness to assume a constructivist culture in their classrooms. The latter point is critical, as it relates back to the holistic view of learning and motivation. Rather than perceive motivation that is done by a teacher to get a student to perform, a constructivist learning culture presupposes the need for students to take ownership of the ideas being learned and strategies for this learning. If teachers' beliefs about the nature and goals of schooling are counter to a constructivist orientation, students should not be expected to derive the benefits of project-based learning.

A good example of more recent constructionist research that has taken such project-based learning factors into account is that of Yasmin Kafai (1994, 1995; Kafai & Harel, 1991). She and her colleagues have conducted a series of studies focused on "children as designers." Their research has explored student motivation and learning while building multimedia projects, usually in the context of students building games and presentations for other, younger, students. In one example (Kafai, Ching, & Marshall, 1997), teams of fifth- and sixth-grade students were asked to build interactive multimedia resources for third graders. This research, predominantly qualitative, investigated how the students approached the task and negotiated their social roles on the team. Interestingly, the students who developed the most screens, or pages, for the team project were not necessarily those who spent the most time on the project or who exhibited the most project leadership. Upon further analysis of individual contributions, it was found that those students who spent the most time on the project focused their efforts more on developing content-related screens and animation, compared to navigational screens. Quantitative data were also included demonstrating that the students' knowledge of astronomy increased significantly as a result of their participation in the project. Research such as this demonstrates that students are able to negotiate successfully the difficult demands of designing and developing multimedia, find the projects to be motivating and relevant, and also gain content knowledge along the way. In a similar example, in which teams of elementary-school students developed computer projects about neuroscience, Kafai and Ching (2001) found that the team-based project approach afforded many unique opportunities for discussions about science *during* the design process. Planning meetings gave students an authentic context in which to engage in systemic discussions about science. Team members who had prior experience in the team project approach often extended these discussions to consider deeper relationships.

A similar project is Project KIDDESIGNER, in which elementary- and middle-school children were asked to take roles on software design teams (Rieber, Luke, & Smith, 1998). The children's goal was to design educational computer games based on content they had just learned in school. The goal of this research was to see whether such a task would be perceived as authentic by the children and to understand how they would perform when given such design tasks in a collaborative context. Game design is both an art and a science—though games, like stories, have well-established parts, the creation of a good game demands much creativity and sensitivity to the audience that will play the games. As an interactive design artifact, it is difficult to evaluate good games just by reading their descriptions and rules. Instead, game prototypes become essential design artifacts for assessing and revising a game's design. Unlike the research by Kafai and her colleagues, the children in Project KIDDESIGNER were not expected to master a programming language, such as Logo, and then program their games. Instead, the children focused exclusively on the design activities, with the researchers acting as their programmers. The results of this study, conducted as a design experiment, showed that the children were able to handle the complexities of the design activity and were able to remain flexible in their team roles. Team members, by and large, were able to negotiate competing solutions to design problems and maintain deadlines. Of particular interest was how the resulting games provided insights into the value the children placed on the school-based content they needed to embed in the games. For example, one of the most popular games used the context of motocross racing where mathematics were embedded as a penalty for poor performance. These children saw mathematics as a punishment for not performing other tasks, which they valued, well.

22.6 MICROWORLDS MORE BROADLY CONCEIVED: GOING BEYOND PROGRAMMING LANGUAGES

Although the roots of microworlds rest in programming languages, or general computational media, such as Logo and Boxer, advances in technology have led to the development of other forms of microworlds, such as those based on direct manipulation of screen objects and attributes. The relative merits of learning text-based programming languages and those that use "point and click" methods of interaction, such as the very popular Geometer's Sketchpad, an icon-based tool for constructing geometric relationships and principles, have been hotly debated (diSessa, Hoyles, Noss, & Edwards, 1995b).

Consider the issue of "curricular fit" of these two types of systems. It is much easier to make the argument for a school to invest in a tool such as Geometer's Sketchpad as compared to Logo or Boxer because Geometer's Sketchpad more readily "maps" on to the current geometry curriculum. diSessa, Hoyles, Noss, and Edwards (1995a) suggest that systems such as Boxer and Logo are usually seen as too "subversive" by mainstream educators, hence their adoption is often resisted, whereas Geometer's Sketchpad fits easily into the curriculum, due to its alignment with traditional curriculum goals. One might argue, then, that the power and affordances of a tool such as Geometer's Sketchpad would be recognized and capitalized on less because many educators would be expected solely to integrate it into the standard way of teaching and learning, hence using it to perpetuate the "standard curriculum," though such use would also improve how the standard curriculum is taught. Another point of view is that a system like Geometer's Sketchpad could be even more subversive than Logo because, once it becomes part of the school system, its affordances may actually help to reconceptualize the boundaries of learning and teaching.

A major factor concerning the widespread adoption of these systems is the belief that each system needs to effect large-scale changes for all learners in a school population. "It is tempting—and prevalent—to attempt to design for the majority; indeed it seems many presume that an encounter with a system will produce some outcome for all. This is, of course, an underlying assumption of schooling: that it is 'good' for all. In fact, exploratory learning environments may have some claim to just the opposite, to be designed for relatively rare occurrences" (diSessa et al., 1995a, pp. 9–10).

22.6.1 ThinkerTools

ThinkerTools (http://thinkertools.soe.berkeley.edu/) is both a computer-based modeling tool for physics and a pedagogy for science education based on scientific inquiry: ". . . an approach to science education that enables sixth graders to learn principles underlying Newtonian mechanics, and to apply them in unfamiliar problem solving contexts. The students' learning is centered around problem solving and experimentation within a set of computer microworlds (i.e., interactive simulations)" (White & Horowitz, 1987, abstract). ThinkerTools is one of the earliest examples of how the concept of a microworld was broadened to go beyond computer programming to include interactions and model building within "interactive simulations."

In the ThinkerTools software, students explore interactive models of Newtonian mechanics. They can build their own models, or they can interact with a variety of ready-made models that accompany the software. A variety of symbolic visual representations is used. Simple objects, in the shape of balls (called "dots"), can be added to the model, each with parameters directly under the student's control. For example, each dot's initial mass, elasticity (bouncy or fragile), or velocity can be manipulated. Variables of the model's environment itself can be modified, such as the presence and strength of gravity and air friction. Other elements can be added to the model, such as barriers and targets. Forces affecting the motion of the balls can be directly controlled, if desired, by the keyboard or a joy stick, such as by giving the ball kicks in the four directions (i.e., up, down, left, right). This adds a video- game-like feature to the model.

The ThinkerTools software also includes a variety of measurement tools with which students can accurately observe distance, time, and velocity. Another symbol, called a datacross, can be used to show graphically the motion variables of the object. A datacross shows the current horizontal and vertical motion of the ball in terms of the sum of all of the forces that have acted on the ball. The motion of the object over time can also be depicted by having the object leave a trail of small, stationary dots. When the object moves slowly, the trail of dots is closely spaced, but when the object moves faster, the space between the trailing dots increases. Students can also use a "step through time" feature, in which the simulation can be frozen in time, allowing students to proceed step by step through time. This gives them a powerful means of analyzing the object's motion and also of predicting the object's future motion. The point of all of these tools is to give students the means of determining and understanding the laws of motion in an interactive, exploratory way: "In this way, such dynamic interactive simulations can provide a transition from students' intuitive ways of reasoning about the world to the more abstract, formal methods that scientists use for representing and reasoning about the behavior of a system" (White & Frederiksen, 2000b, pp. 326–327). Similar to Papert's idea of a transitional object, the ThinkerTools software acts as a bridge between concrete, qualitative reasoning of real-world examples and the highly abstract world of scientific formalism where laws are expressed mathematically in the form of equations.

The ThinkerTools software is best used, according to White, with an instructional approach to inquiry and modeling called the ThinkerTools Inquiry Curriculum. The goal of this curriculum is to develop students' metacognitive knowledge, that is, "their knowledge about the nature of scientific laws and models, their knowledge about the processes of modeling and inquiry, and their ability to monitor and reflect on these processes so they can improve them" (White & Frederiksen, 2000b, p. 327). White and her colleagues predicted that such a pedagogical approach used in the context of powerful tools such as the ThinkerTools software should make learning science possible for all students. The curriculum largely follows the scientific method, involving the following steps: (1) question—students start by constructing a research question, perhaps the hardest part of the model; (2) hypothesize—students generate hypotheses related to their question; (3) investigate—students carry out experiments, both with the ThinkerTools software and in the real world, the goal of which is to gather empirical evidence about which hypotheses (if any) are accurate; (4) analyze—after the experiments are run, students analyze the resulting data; (5) model—based on their analysis, students articulate a causal model, in the form of a scientific law, to explain the findings; and (6) evaluate—the final step is to test whether their laws and causal models work well in real-world situations, which, in turn, often leads to new research questions.

White and Frederiksen (2000b) also reported interesting insights into how teachers using ThinkerTools can affect the learning outcomes of the materials. For example, they describe teachers who contacted them to use their materials, teachers with whom they were not already associated. Eight such teachers were asked to administer the physics and inquiry tests that come with the materials and send the results back to White. Interestingly, four of the teachers reported that their focus was on using ThinkerTools as a way to teach physics. The students of these teachers showed a significant improvement on the physics test but not on the inquiry test. In contrast, the other four teachers said that their focus was on teaching scientific inquiry—their students improved significantly on both their inquiry *and* their physics expertise. Obviously, the goals of the teacher can lead to many missed opportunities for inquiry learning.

22.6.2 SimCalc

The SimCalc project (http://www.simcalc.umassd.edu/) is concerned with the mathematics of change and variation (MCV). Its mission is to give ordinary children the opportunities, experiences, and resources they need to develop an extraordinary understanding of and skill with MCV (Roschelle et al., 2000). The SimCalc project is based on three lines of innovation. The first is a deep reconstruction of the calculus curriculum, both its subject matter and the way in which it is taught. The goal is to allow all children, even those in elementary school, to access the mathematical principles of change and variation. The developers assert that this is possible through the design of visualizations and simulations for collaborative inquiry. The most notable innovation in the SimCalc curriculum is the use of piecewise linear functions as the basis of student exploration. In a velocity graph, for example, a student can build a function by putting together line segments, each of the same time duration. A series of joined horizontal segments denotes constant velocity and a set of rising or falling segments denotes increasing or decreasing speed. The second innovation is to root the learning of these mathematics principles in meaningful experiences of students. Students bring with them a wealth of mathematical understanding that is largely untapped in traditional methods of learning calculus. The SimCalc project does not require students to understand algebra before exploring calculus principles. The third innovation is the creative use of technology, namely, special software called MathWorlds.

The MathWorlds software makes extensive use of concrete visual representations, coupled with graphs that students can directly manipulate and control. The graphs can be based on data sets generated by computer-based simulations (animated clowns, ducks, and elevators), laboratory experiments, and even the students' own body movements by capturing their movements with microcomputer-based (or calculator-based) motion sensors, then importing these data into the computer.

Although mathematics educators have spent much time and effort reforming the calculus curriculum, the SimCalc project differs in two important ways from these efforts. First, unlike traditional reform, which has focused solely on the teaching of calculus in high school, the SimCalc project has reconceptualized the teaching of mathematics at all grade levels, starting with elementary school. Second, other reform efforts have focused on linking numeric, graphic, and symbolic representations, whereas the SimCalc project has put its focus on meaningful student experience based on graphs of interesting visual phenomena that students can manipulate directly. The SimCalc project places much value on students experiencing phenomena as the basis for their mathematical explorations.

The SimCalc curriculum is based on four strategies that counter traditional teaching of calculus. First, phenomena are studied and understood before delving into mathematical formalisms. Second, the mathematics are based on discrete variation before turning to continuous variation. Third, the mathematics of accumulation and integrals are taught before rates of change and derivatives. Fourth, students learn to master graphs before algebraic symbolism. So, instead of requiring algebra as a prerequisite skill for studying calculus, the SimCalc project using students' grasp of visual problem solving with graphs to enter the mathematical world of change and varying quantities.

Research with SimCalc since the project began in about 1993 has focused on two themes. The first research phase investigated the use of the MathWorlds software on student cognition, technology designs, and alternative curricular sequences. This effort resulted in a "proof of concept" curriculum largely divorced from systemic educational factors. Again, much of the research in this phase can be characterized as design experiments. The second research phase, just beginning, has focused specifically on such systemic issues as curricular integration, teacher professional development, and assessment.

Early SimCalc research was characterized as large field-test trials designed to generate formative data to improve the software and refine the SimCalc curricular approach. Although less rigorously implemented than experimental research, data from these early field trials demonstrated that the seventh-, eighth-, and ninth-grade students who participated in the SimCalc curriculum significantly improved in their understanding of rate of change problems. Interestingly, although these formative data show that middle-school students can effectively solve mathematical problems involving change and variation, the exciting possibility of introducing younger students to these principles is greatly hampered by the fact that calculus is taught only as part of the high-school curriculum. This content is considered an "add-on" to an already full middle-school mathematics curriculum. Ironically, despite the exciting potential that students could have access to such powerful mathematical ideas at a younger age, these learning opportunities are largely resisted by schools due to the curriculum constraints. Fortunately, these obstacles are exactly those that the SimCalc team hopes to study in the next phase of the project.

22.6.3 GenScope

Genscope (http://genscope.concord.org/) is an exploratory software environment "designed to help students learn to reason and solve problems in the domain of genetics" (Horwitz & Christie, 2000, p. 163). The goal of GenScope is to help students understand scientific explanations and also to gain insight into the nature of the scientific process. Horwitz and his colleagues describe GenScope as a "computer-based manipulative" and insist that it is neither a simulation nor a modeling tool. Interestingly, their intent is to have students use it to try to determine, largely through inductive reasoning, the software's underlying model (i.e., genetics). This is precisely the aim of much research on educational uses of simulations. Like other microworlds, the emphasis of GenScope is on qualitative understanding of the domain. It gives students a way to represent genetic problems and derive solutions interactively. It does not require students to master the vocabulary of genetics before effectively using genetic concepts and principles. Indeed, Horwitz and his colleagues suggest that traditional science instruction poses a significant linguistic barrier to understanding genetics—typical science textbooks often introduce more vocabulary per page than do foreign language texts. This linguistic barrier is compounded by the fact that the science terms usually do not have a direct analogue in the student's "first language" and, hence, are actually more difficult to learn than a foreign language.

Another significant barrier in understanding genetics, according to Horwitz, is the mismatch between how scientists actually study genetics and how it is taught. Understanding genetics is largely an inductive exercise, trying to determine the cause from an observed set of effects. In contrast, most science teaching is deductive, teaching the rule, followed by students having to deduce the results. Moreover, the skills that a scientist uses are rarely taught in the classroom (i.e., using the scientific method to reason inductively). Instead, most classroom practice activities are meant to let students rehearse factual information and solve similar problem sets. Of course, knowing a correct answer on a worksheet does not mean that a student actually understands the underlying concepts and principles. The GenScope curriculum was designed to have students use the GenScope tool in ways that mirror closely the methods used by actual scientists.

Genetics is the study of how an organism inherits physical characteristics from its ancestors and passes them on to its descendants, the rules of which were first postulated by Gregor Mendel in the 1800s. Learning genetics is particularly challenging because descriptions of how changes occur can be formulated at many different levels. GenScope provides students with six interdependent levels: molecules, chromosomes, cells, organisms, pedigrees, and populations. GenScope provides students with a simplified model of genetics for them to manipulate, beginning with the imaginary species of dragons. GenScope provides individual computer windows for each of the levels—students can interact with one of the levels, say via a DNA window to show the genes of an organism (i.e., genes that control whether a dragon has wings), and then see the results of their manipulation in the organism window (i.e., a dragon sprouting wings).

22.6.4 Pedagogical Approach of GenScope

Students using GenScope start by focusing on the relationships between the organism and the chromosome levels using the fictitious dragon species, progressively working up to higher levels of relationships dealing with real animals. After getting familiar with the GenScope interface for a few minutes, students are immediately given a challenge (e.g., a fire-breathing green dragon with legs, horns, and a tail but no wings). Students quickly master the ability to manipulate the genes at the chromosomal level to produce such an animal. Interestingly, the next step is to switch to a paper-and-pencil activity where students are asked to describe what a dragon would look like given printed screen shots of chromosomes. After students construct an answer, they are encouraged to use GenScope to verify, or correct, their answers. Students then progress to interrelating the DNA level to the chromosome and organism level. Students come to learn about how recessive and dominant genes can be combined to produce certain characteristics. For example, if wings are a recessive trait, a dragon would have to possess two recessive genes to be born with wings. Students then progress to the cell level and consider how two parents may pass traits to their offspring. As shown, the pedagogical approach used here is to challenge students with problems to solve in GenScope, then give them time to work alone or in pairs to solve the problems through experimentation.

A variety of research with GenScope has been conducted to test the hypothesis that students using GenScope would be better able to demonstrate genetic reasons among multiple levels than students not using GenScope. An early study compared one class of students using GenScope to another using a traditional textbook-based curriculum. Interestingly, although the GenScope students definitely showed greater qualitative reasoning as evidenced in the observations of their computer interactions, they were unable to outperform the other students on traditional paper-and-pencil tests. Horwitz and his colleagues explain these early results in several ways. First, and not surprisingly, this type of media comparison research does not lead to equal comparisons. Students were not learning the same content in similar ways or at similar rates. While, on one hand, the GenScope group was asked to solve richer and more sophisticated problems than the other group, they were doing so through the interactive and successive manipulation possible with GenScope. The textbook group was forced throughout to use genetic formalism, such as the vocabulary found on the tests (e.g., phenotype, genotype, allele, meiosis, heterozygous, homozygous, dominant/recessive).

Besides the language barrier that GenScope students faced, Horwitz and his colleagues suggest three other barriers that serve to prevent students using microworlds like GenScope to demonstrate their increased understanding on most traditional tests. The first, "shift in modality," is the barrier between shifting from computer interactions to paper-and-pencil ones. The second, "examination effect," argues that the very act of taking a test negatively affects student performance. The final barrier concerns the fact that any understanding learned in context is qualitatively different from understanding gained through abstract symbols, such as the written word. In sum, students have a very difficult time translating their GenScope-based understanding of genetics to performance on traditional paper-and-pencil tests. Horwitz and his colleagues accept this challenge given that such measures are part of the political reality of arguing for using new technologies and curricula within schools.

Other research shows that students in general biology and general science classes show much larger gains in their understanding of genetics than students in college-prep or honors biology. The larger gains were particularly evident in classrooms where teachers used curricular materials especially designed to scaffold aspects of their learning of genetics (Hickey, Kindfield, & Wolfe, 1999).

22.7 THEORETICAL BASIS FOR LEARNING IN A MICROWORLD

Based on the examples considered in this chapter, microworlds are clearly an eclectic and varied assortment of software and pedagogical approaches to learning. Is there a clear theoretical basis for suggesting that microworlds offer a more powerful representation for problem- solving with domains such as physics and mathematics? Perkins and Unger (1994) suggest that their power resides in the way microworlds represent a problem for the student. We use all sorts of representations to understand and solve problems. However, the teaching of certain domains, most notably science and mathematics, has tended to use technical representations (e.g., algebra, equations, and graphs) rather than less technical representations (e.g., analogies, metaphors, and stories).

Domains such as mathematics and physics have been represented in a variety of ways to discover the boundaries and underlying laws of the domain. The ways in which people such as Galileo, Newton, Einstein, and Feynman have chosen to represent the field of physics is a useful historical review of representation.

What role do representations play in understanding? Using the classic problem of why falling objects of different weights fall at the same rate, Perkins and Unger (1994) offer three complementary approaches using very different representations: algebraic, qualitative, and imagistic. An algebraic approach uses mathematical manipulation of the relevant formulas (e.g., Newton's second law of motion, or force equals mass times acceleration) to explain the result. In a qualitative explanation, one could reason that the greater downward force expected on a larger mass would be equally offset by the fact that a larger mass is also harder to "get moving." Finally, an imagistic explanation involves a kind of "thought experiment," such as that actually described by Galileo, to reason through the problem. For example, Galileo imagined the motion of two iron balls connected by a metal rod and how the motion would change as the balls fell if the connecting rod were made thinner and thinner, eventually being connected with just a thin thread, and then, finally, imagined the thread being cut while the balls were falling. From such reasoning through imagery, it is clear that the acceleration of the balls would not vary regardless of their mass.

Perkins and Unger (1994) suggest that microworlds offer a fourth and different kind of representation. They argue that representation facilitates explanation through active problem solving, similar to the search that a user executes in a "problem space" proposed by Newell and Simon (1972). Such a search involves an initial state, a goal state, various intermediate states, and operations that take the student from one state to another. The objective is to turn the initial state into the goal state. How to search the problem space for a path to the solution depends on a variety of factors, such as the student's knowledge of the domain in which the problem is situated (e.g., physics), the student's general abilities, and the way the problem space is represented for the student. To say that a student *understands* a problem is to mean, according to Perkins and Unger (1994), that he or she can perform the necessary explanation, justification, and prediction related to the problem topic. (They use the term *epistemic problems* to describe these sorts of problem-solving performances.)

Representations aid problem solving in three ways. First, the right representation reduces the cognitive load and allows students to use their precious working memory for higher-order tasks. For example, algebra uses symbols that are very concise and uses rules that are very generalizable to a range of problems. Of course, this is true only when the students have already mastered algebra. Qualitative representations, such as those based on analogies and metaphors, allow students to think of a problem first in terms of an example already known, such as the idea of electricity being like water in a pipe. Second, representations clarify the problem space for students, such as by organizing the problem and the search path. Again, the rules of algebra offer beginning, middle, and end states to reach and clear means of transforming equations to these different states. Qualitative representations offer the user models to use and compare. Similarly, imagistic representations help to reveal a critical factor in solving the problem, such as the absurd role played by the silk thread in Galileo's thought experiment. Third, a good representation reveals immediate implications. Regardless of how well a representation may minimize the cognitive load or clarify the problem space, if students do not see immediate applications while engaged in the problem search, then the solutions found will be devoid of meaning for, and hence understanding by, the students.

Microworlds offer the means of maximizing all three benefits of representations, when used in the context of an appropriate science teaching pedagogy, such as one based on the scientific method of hypothesis generating and hypothesis testing. For example, in the ThinkerTools microworld, students directly interact with a dynamic object while having the discrete forces they impart on the object horizontally or vertically displayed on a simple, yet effective datacross. Students can also manipulate various parameters in the microworld, such as gravity and friction. ThinkerTools ably creates a problem space in which numeric, qualitative, and visual representations consistently work together.

Not only do computer-based microworlds afford reducing the cognitive load, clarifying the problem space, and revealing immediate implications, but also, Perkins and Unger (1994) go on to suggest, microworlds afford the integration of structure-mapping frameworks based on analogies and metaphors. Similarly, a microworld can be designed so as to provide a representation that purposefully directs a student to focus on the most salient relationships of the phenomena being studied. Of course, such benefits do not come without certain costs or risks. For example, as with the use of any analogy, if the users do not correctly understand the mapping structure of the analogy, then the benefits will be lost and the students may potentially form misconceptions. The danger of a microworld's misleading students if they do not understand the structural mappings well is real. Just providing a microworld to students, without the pedagogical underpinnings, should not be expected to lead to learning. The role of the teacher and the resulting classroom practice is crucial here. Microworlds rely on a culture of learning in which students are expected to inquire, test, and justify their understanding. "Students needs to be actively engaged in the construction and assessment of their understandings by working thoughtfully in challenging and reflective problem contexts" (p. 27). (See pages 27–29 for more risks and pitfalls.)

As Perkins and Unger (1994) point out, microworld designers have a formidable task; they

have to articulate adequately the components and relationships among components of the domain to be learned. Next the designers have to construct an illustrative world exemplifying that targeted domain. Finally, the illustrative world should provide natural or familiar referents that, when placed in correspondence with one another and mapped to the target domain, yield a better understanding of the domain. (p. 30)

22.8 THE RELATIONSHIP AMONG MICROWORLDS, SIMULATIONS, AND MODELING TOOLS

There are many other examples of innovative software applications that are usually clumped in the microworld camp, the most notable being Geometer's Sketchpad (Olive, 1998). There is also the range of modeling packages to consider, such as Interactive Physics and Stella, and simulations such as SimCity. Should these be classified as microworlds? Determining the answer depends on how the user appropriates the tool using the five microworld attributes discussed earlier in this chapter. However, despite the controversy between giving users programmable media (i.e., Logo and Boxer) and giving them preprogrammed models of systems, there do seem to be benefits to including an analysis of modeling tools and simulations in a discussion of microworlds (Rieber, 1996).

There are two main ways to use simulations in education: model using and model building. Model using is when you learn from a simulation designed by someone else. This is common of *instructional* approaches where simulations are used as an interactive strategy or event, such as practice. Learning from using a simulated model of a system is different from learning from building working models in that the student does not have access to the programming of the simulation. The student is limited to manipulating only the parameters or variables that the designer of the simulation embedded into the simulation's interface. For example, in a simulation of Newtonian motion the user may have only the ability to change the mass of an object in certain increments, and not the ability to change the initial starting positions of the objects or even how many objects will interact when the simulation is run. In contrast, in model building the learner has a direct role in the construction of the simulation. This approach is closely related to work with microworlds.

The question of when a microworld is or is not a simulation often troubles people. While ThinkerTools or Interactive Physics displays trajectories of simulated falling balls, the underlying mathematical model makes the resulting representation much more "real" than a paper-and-pencil model. And although the ability to stop a ball in midflight has no analogue in the real world, features like this make understanding the real world more likely. What is important is that the mathematical models of these environments represent the phenomenon or concept in question accurately, followed by exploiting the representation for educational purposes. However, a tool like Geometer's Sketchpad is clearly *not* a simulation—its geometry is as real as it gets.

The model-using approach to simulations has had a long history in instructional technology, particularly in corporate and military settings. However, simulations have become very popular designs in the education market. There are three major design components to an educational simulation: the underlying model, the simulation's scenario, and the simulation's instructional overlay (Reigeluth & Schwartz, 1989). The underlying model refers to the mathematical relationships of the phenomenon being simulated. The scenario provides a context for the simulation, such as space travel or sports. The instructional overlay includes any features, options, or information presented before, during, or after the simulation to help the user explicitly identify and learn the relationships being modeled in the simulation. The structure and scope of the instructional overlay are of course, an interesting design question and one that has shaped my research. Mental model theory offers much guidance in the design of an effective scenario and instructional overlay, such as thinking of them as an interactive conceptual model (Gentner & Stevens, 1983; Norman, 1988). This supports the idea of using metaphors to help people interact with the simulation (Petrie & Oshlag, 1993).

de Jong and van Joolingen (1998) present one of the most thorough reviews of scientific discovery learning within computer-based simulations (of the model-using type). The goal of this type of research is to present a simulation to students and ask them to infer the underlying model on which the simulation is based. Scientific discovery learning is based on a cycle corresponding to the steps of scientific reasoning: defining a problem, stating a hypothesis about the problem, designing an experiment to test the hypothesis, collecting and analyzing data from the experiment, making predictions based on the results, and making conclusions about and possible revisions of the robustness of the original hypotheses.

The research reviewed by de Jong and van Joolingen (1998) shows that students find it difficult to learn from simulations using discovery methods and need much support to do so successfully. Research shows that students have difficulty throughout the discovery learning process. For example, students find it difficult to state or construct hypotheses that lead to good experiments. Furthermore, students do not easily adapt hypotheses on the basis of the data collected. That is, they often retain a hypothesis even when the data they collect disconfirm the hypothesis. Students do not design appropriate experiments to give them pertinent data to evaluate their hypotheses. Students are prone to *confirmation bias,* that is, they often design experiments that will lead to support their hypotheses. Students also find interpreting data in light of their hypotheses to be very challenging. In light of these difficulties de Jong and van Joolingen (1998) also review research on ways to mitigate these difficulties. One conclusion they draw is that information or instructional support needs to come *while* students are involved in the simulation, rather than prior to their working with the simulation. That is, students are likely to benefit from such instructional interventions when they are confronted with the task or challenge. This often flies in the face of conventional wisdom that students should be prepared thoroughly before being given access to the simulation. The research also shows that embedding guided activities within the simulation, such as exercises, questions, and even games, helps students to learn from the simulation. When designing experiments, students can benefit from experimentation hints, such as the recommendation to change only one variable at a time. de Jong and van Joolingen (1998) also conclude that the technique of *model progression* can be an effective design strategy. Instead of presenting the entire simulation to students from the onset, initially students are given a simplified version, then variables are added as their understanding unfolds. For example, a Newtonian simulation could be presented first with only one-dimensional motion represented, then with two-dimensional motion.

Finally, de Jong and van Joolingen (1998) also point out the importance of understanding how learning was measured in a particular study. There is a belief that learning from simulations leads to "deeper" cognitive processing than learning from expository methods (such as presentations). However, many studies did not test for application and transfer, so it is an open question whether a student who successfully learns only how to manipulate the simulation can apply this knowledge to other contexts. A student who successfully manipulates the simulation may not have acquired the general conceptual knowledge to succeed at other tasks. The review by de Jong and van Joolingen shows that there is still much researchers need to learn about the role of simulations in discovery learning and, also, about how to design supports and structure to help students use the affordances of simulations most effectively. There are also many styles and strategies beyond scientific discovery learning. For example, an experiential or inductive approach would have students explore a simulation first, followed by providing organized instruction on the concepts or principles modeled by the simulation. With this approach, the simulation provides an experiential context for anchoring later instruction.

22.9 CONCLUSION

Microworlds describe both a class of interactive exploratory software and a particular learning style. This chapter has taken a close look at the software, philosophy, and research of some of the most prominent and successful microworlds developed since about 1980—Logo, Boxer, ThinkerTools, SimCalc, and Genscope. All are incredibly creative and powerful, and all fully capture the interactive and computational affordances of computers for exploratory learning. The microworlds described in this chapter are but a few of those developed. There are many others that deserve notice, such as Mitchell Resnick's (1991, 1994, 1996, 1999) StarLogo (http://education.mit.edu/starlogo/), a version of Logo that allows thousands of turtles to be active at the same time and all under the control of the user through a few simple commands. This powerful computational medium gives children a doorway to the world of decentralized systems, which include such complex phenomena as traffic jams, ant colonies, and even the migration of birds. Unfortunately, insufficient research is yet available on this provocative computational medium. Sadly, this reflects the fact that there is less research in the microworld literature than one would expect and hope providing evidence of their use and impact in the schools.

In the case of microworlds derived from computational media, such as Logo and Boxer, hundreds of even smaller microworlds have been developed as individual programs, though they remain open to change by the user. Probably the most successful microworld of the past 25 years has been turtle geometry, a subset of the original capabilities of the Logo language and a continuing part of many other languages (including Boxer) and programs. It is conservative to state that tens of thousands of children have successfully learned to control the turtle to make interesting geometric shapes. Most of these children, regrettably, never progressed to the higher levels of programming possible with these languages or even within the turtle geometry microworld itself. Explanations of this are speculative, the most likely being that the educational system has yet to adopt a true constructivist perspective. Although curricula in math and science supported by the respective professional associations have repeatedly called for increased attention to problem solving and scientific inquiry, most school curricula are still based on getting all students through all topics at about the same time. Until the focus turns from "covering the material" to student meaning making, it is unlikely that any microworld, no matter how powerful or persuasive, will have much influence on student learning. As David Perkins (1986) points out,

> . . . Fostering transfer takes time, because it involves doing something special, something extra. With curricula crowded already and school hours a precious resource, it is hard to face the notion that topics need more time than they might otherwise get just to promote transfer. Yet that is the reality. It is actually preferable to cover somewhat less material, investing the time thereby freed to foster the transfer of that material, than to cover somewhat more and leave it context-bound. After all, who needs context-bound knowledge that shows itself only within the confines of a particular class period, a certain final essay, a term's final exam? In the long haul, there is no point to such instruction. (p. 229)

While microworld development over the past 25 years has been impressive, there is an urgent need to launch aggressive research programs so that the potential of these programs is not demonstrated in but a few special classrooms that get the chance to participate in field trials complete with able university personnel who come in to ensure that wonderful things will happen. Interestingly, most of the serious research on these systems has been completed by Ph.D. students at schools such as MIT, the University of California, Berkeley, and Harvard University for their doctoral dissertations. Some, such as the research

TABLE 22.1. Partial List of Doctoral Dissertation Research Project Microworlds

Advisor	Dissertation Title, Ph.D. Candidate, Year	Microworld
Seymour Papert	Twenty Heads Are Better Than One: Communities of children as Virtual Experts Michele Joelle Pezet Evard 1998	Logo
Seymour Papert	They Have Their Own Thoughts: Children's Learning of Computational Ideas from a Cultural Constructionist Perspective Paula K. Hooper 1998	Logo
Seymour Papert	Expressive Mathematics: Learning by Design David W. Shaffer 1998	Geometer's Sketchpad
Seymour Papert	Connected Mathematics: Building Concrete Relationships with, Mathematical Knowledge Uri J. Wilensky 1993	Logo
Seymour Papert	Beyond the Centralized Mindset: Explorations in Massively-Parallel, Microworlds Mitchel Resnick 1992	StarLogo
Seymour Papert	Learning Constellations: A Multimedia Ethnographic Research, Environment Using Video Technology for Exploring Children's Thinking (Ethnography) Ricki Goldman Segall 1990	Logo
Andy diSessa	Student Control of Whole-Class Discussions in a Community of Designers Peter Birns Atkins Kindfield 1996	Boxer
Andy diSessa	The Symbolic Basis of Physical Intuition: A Study of Two Symbol Systems in Physics Instruction Bruce L. Sherin 1996	Boxer
Andy diSessa	Students' Construction of Qualitative Physics Knowledge: Learning about Velocity and Acceleration in a Computer Microworld (Physics Education) Jeremy M. Roschelle 1991	Envisioning Machine
Andy diSessa	Learning Rational Number (Constructivism) John P. Smith, III 1990	Boxer
David Perkins	Minds in Play: Computer Game Design as a Context for Children's Learning (Vol. I and II) Yasmin B. Kafai 1993	Logo
Barbara White	Student Goal Orientation in Learning Inquiry Skills with Modifiable Software Advisors Todd A. Shimoda 1999	ThinkerTools
Barbara White	Developing Students' Understanding of Scientific Modeling Christine V. Schwarz 1998	ThinkerTools

by Barbara White, Idit Harel, and Yasmin Kafai, we have already presented. There is more, such as Jeremy Roschelle's (1991) early research on a physics microworld called the Envisioning Machine, which led to his collaborative work on MathWorlds in the SimCalc project. Table 22.1 lists a few notable examples of doctoral research carried out as part of microworld efforts.

Following in the footsteps of Papert, all of the microworld developers write persuasively about their software and pedagogical approaches. Their writings are provocative, challenging, and oftentimes inspiring. They all have interesting stories to tell about the field tests with their software. Among the lessons learned from these stories is that the potential of the software to make a difference in a child's access and understanding of complex domains, such as geometry, calculus, physics, and genetics, is great. But the challenges leading to such learning, based on constructivist orientations, are formidable. The educational system needs to change in fairly dramatic ways for the potential of these systems to be realized. Probably the most fundamental change is allowing students adequate time, coupled with providing a master teacher who not only knows the software well, but also is a master of constructivist teaching—someone who knows how and when to challenge, provoke, suggest, scaffold, guide, direct, teach, and, most of all, leave a group of students alone to wrestle with a problem on their own terms. The word "facilitate" is often used ambiguously to denote such a teacher's actions. Such a role elevates the teacher's status and importance in the classroom, and although it can lead to a more satisfying form of teaching, it is a difficult style to master.

Without question, microworlds are among the most creative developments within educational computing and the learning sciences. Though all are defined as exploratory learning environments, all are also goal-oriented to some extent. This implies that microworlds offer a way to bridge the gap between the objectivism of instructional design methods and constructivist notions of learning. In other words, because the boundaries of a microworld are designed with certain constraints that lead and help learners to focus on a relatively narrow set of concepts and principles, microworlds complement any instructional system that requires the use of and accounting for predetermined instructional objectives (Rieber, 1992). This is not to say that conflicts do not exist. Indeed, the inability or unwillingness of schools to allow teachers and students to devote adequate time to inquiry-based activities using microworlds due to curriculum demands is a case in point. Yet, as constructivist perspectives aligned with technology innovations mature, as evidenced by the many microworld projects discussed in this chapter, there is hope that the long-rival factions within constructivist and instructivist "camps" will continue to realize more that they have in common. The current interest in and maturity of design experiments offer great promise in stimulating much more microworld research that will also be rigorously and authentically assessed.

References

Abelson, H. (1982). Logo for the Apple II. Peterborough. NH: BYTE/McGraw Hill.

Adams, S. T., & diSessa, A. (1991). Learning by "cheating": Students' inventive ways of using a boxer motion microworld. *Journal of Mathematical Behavior, 10*(1), 79–89.

Barab, S. A., & Kirshner, D. (2001). Guest editors' introduction: Rethinking methodology in the learning sciences. *Journal of the Learning Sciences, 10,* 5–15.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palinscar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist, 26*(3 & 4), 369–398.

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

Chi, M., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5,* 121–152.

Clark, R. E. (1994). Media will never influence learning. *Educational Technology Research & Development, 42*(2), 21–29.

Clark, R. E. (Ed.). (2001). *Learning from media: Arguments, analysis, and evidence*. Greenwich, CT: Information Age.

Clements, D. (1989). *Computers in elementary mathematics education*. Englewood Cliffs, NJ: Prentice Hall.

Clements, D. H. (1984). Training effects on the development and generalization of Piagetian logical operations and knowledge of number. *Journal of Educational Psychology, 76,* 766–776.

Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology, 78,* 309–318.

Clements, D. H. (1987). Longitudinal study of the effects of Logo programming on cognitive abilities and achievement. *Journal of Educational Computing Research, 3,* 73–94.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology, 76*(6), 1051–1058.

Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 15–22). New York: Springer-Verlag.

Cuban, L. (1986). *Teachers and machines: The classroom of technology since 1920*. New York: Teachers College Press.

Cuban, L. (2001). *Oversold and underused: Computers in the classroom*. Cambridge, MA: Harvard University Press.

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*(2), 179–201.

Dewey, J. (1916). *Democracy and education: An introduction to the philosophy of education*. New York: Macmillan.

diSessa, A. A. (1989). *Computational media as a foundation for new learning cultures.* Technical Report G5. Berkeley: University of California.

diSessa, A. A. (1997). Twenty reasons why your should use Boxer (instead of Logo). In M. Turcsányi-Szabó (Ed.), *Learning & Exploring with Logo: Proceedings of the Sixth European Logo Conference, Budapest, Hungary* (pp. 7–27).

diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge, MA: MIT Press.

diSessa, A. A., Abelson, H., & Ploger, D. (1991). An overview of Boxer. *Journal of Mathematical Behavior, 10,* 3–15.

diSessa, A. A., Hoyles, C., Noss, R., & Edwards, L. D. (1995a). Computers and exploratory learning: Setting the scene. In A. A. diSessa, C. Hoyles, R. Noss, & L. D. Edwards (Eds.), *Computers and exploratory learning* (pp. 1–12). New York: Springer.

diSessa, A. A., Hoyles, C., Noss, R., & Edwards, L. D. (Eds.). (1995b). *Computers and exploratory learning*. New York: Springer.

Eccles, J. S., & Wigfield, A. (1995). In the mind of the actor: The structure of adolescents' achievement task values and expectancy-related beliefs. *Personality and Social Psychology Bulletin, 21,* 215–225.

Edelson, D. C. (2002). Design research: What we learn when we engage in design. *Journal of the Learning Sciences, 11,* 105–121.

Edwards, L. D. (1995). Microworlds as representations. In A. A. diSessa, C. Hoyles, R. Noss, & L. D. Edwards (Eds.), *Computers and exploratory learning* (pp. 127–154). New York: Springer.

Feurzeig, W. (1999). A visual modeling tool for mathematics experiment and inquiry. In W. Feurzeig & N. Roberts (Eds.), *Modeling and simulation in science and mathematics education* (pp. 95–113). New York: Springer-Verlag.

Feurzeig, W., & Roberts, N. (1999). Introduction. In W. Feurzeig & N. Roberts (Eds.), *Modeling and simulation in science and mathematics education* (pp. xv–xviii). New York: Springer-Verlag.

Forrester, J. W. (1989). The beginning of system dynamics. *International meeting of the System Dynamics Society, Stuttgart, Germany* [online]. Available: http://sysdyn.mit.edu/sdep/papers/D-4165-1.pdf.

Gentner, D., & Stevens, A. (Eds.). (1983). *Mental models*. Mahwah, NJ: Lawrence Erlbaum Associates.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments, 1,* 1–32.

Harel, I., & Papert, S. (1991). Software design as a learning environment. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 41–84). Norwood, NJ: Ablex.

Hickey, D. T., Kindfield, A. C. H., & Wolfe, E. W. (1999, April). *Assessment-oriented scaffolding of student and teacher performance in a technology-supported genetics environment*. Paper presented at the annual meeting of the American Educational Research Association, Montreal, Quebec, Canada.

Horwitz, P., & Christie, M. A. (2000). Computer-based manipulatives for teaching scientific reasoning: An example. In M. J. Jacobson & R. B. Kozma (Eds.), *Learning the sciences of the 21st century: Research, design, and implementing advanced technology learning environments* (pp. 163–191). Mahwah, NJ: Lawrence Erlbaum Associates.

Horwitz, P., & Christie, M. A. (2002, April). *Hypermodels: Embedding curriculum and assessment in computer-based manipulatives*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

Jackson, S., Stratford, S. J., Krajcik, J. S., & Soloway, E. (1996). Making dynamic modeling accessible to pre-college science students. *Interactive Learning Environments, 4*(3), 233–257.

Jonassen, D. (1991a). Hypertext as instructional design. *Educational Technology Research & Development, 39*(1), 83–92.

Jonassen, D. (1991b). Objectivism versus constructivism: Do we need a new philosophical paradigm? *Educational Technology Research & Development, 39*(3), 5–14.

Jonassen, D. H. (1992). Designing hypertext for learning. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 123–131). New York: Springer-Verlag.

Jonassen, D. H. (1996). *Computers in the classroom: Mindtools for critical thinking*. Upper Saddle River, NJ: Prentice Hall.

Kafai, Y. (1994). Electronic play worlds: Children's construction of video games. In Y. Kafai & M. Resnick (Eds.), *Constructionism in practice: Rethinking the roles of technology in learning*. Mahwah, NJ: Lawrence Erlbaum Associates.

Kafai, Y. (1995). *Minds in play: Computer game design as a context for children's learning*. Mahwah, NJ: Lawrence Erlbaum Associates.

Kafai, Y., & Harel, I. (1991). Learning through design and teaching: Exploring social and collaborative aspects of constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 85–106). Norwood, NJ: Ablex.

Kafai, Y. B., & Ching, C. C. (2001). Affordances of collaborative software design planning for elementary students' science talk. *Journal of the Learning Sciences, 10*(3), 323–363.

Kafai, Y. B., Ching, C. C., & Marshall, S. (1997). Children as designers of educational multimedia software. *Computers and Education, 29,* 117–126.

Kozma, R. B. (1994). *Will* media influence learning? Reframing the debate. *Educational Technology Research & Development, 42*(2), 7–19.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Upper Saddle River, NJ: Prentice Hall.

Newman, D. (1990). Opportunities for research on the organizational impact of school computers. *Educational Researcher, 19*(3), 8–13.

Newman, D. (1992). Formative experiments on the coevolution of technology and the educational environment. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 61–70). New York: Springer-Verlag.

Norman, D. A. (1988). *The psychology of everyday things*. New York: Basic Books.

Norman, D. A. (1993). *Things that make us smart: Defending human attributes in the age of the machine*. Reading, MA: Addison–Wesley.

Ogborn, J. (1999). Modeling clay for thinking and learning. In W. Feurzeig & N. Roberts (Eds.), *Modeling and simulation in science and mathematics education* (pp. 5–37). New York: Springer-Verlag.

Olive, J. (1998). Opportunities to explore and integrate mathematics with "The Geometer's Sketchpad." In R. Lehrer & D. Chazan (Eds.), *Designing learning environments for developing understanding of geometry and space* (pp. 395–418). Mahwah, NJ: Lawrence Erlbaum Associates.

Papert, S. (1980a). Computer-based microworlds as incubators for powerful ideas. In R. Taylor (Ed.), *The computer in the school: Tutor, tool, tutee* (pp. 203–210). New York: Teacher's College Press.

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

Papert, S. (1987). Computer criticism vs. technocentric thinking. *Educational Researcher, 16*(1), 22–30.

Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–11). Norwood, NJ: Ablex.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York: Basic Books.

Pea, R., & Kurland, M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology, 2,* 1137–1168.

Penner, D. E. (2000/2001). Cognition, computers, and synthetic science: Building knowledge and meaning through modeling. *Review of Research in Education, 25,* 1–35.

Perkins, D. N. (1986). *Knowledge as design*. Mahwah, NJ: Lawrence Erlbaum Associates.

Perkins, D. N., & Unger, C. (1994). A new look in representations for mathematics and science learning. *Instructional Science, 22,* 1–37.

Petrie, H. G., & Oshlag, R. S. (1993). Metaphor and learning. In A. Ortony (Ed.), *Metaphor and thought* (2nd ed., pp. 579–609). Cambridge: Cambridge University Press.

Reigeluth, C., & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. *Journal of Computer-Based Instruction, 16*(1), 1–10.

Resnick, M. (1991). Overcoming the centralized mindset: Towards an understanding of emergent phenomena. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 204–214). Norwood, NJ: Ablex.

Resnick, M. (1994). *Turtles, termites, and traffic jams*. Cambridge, MA: MIT Press.

Resnick, M. (1996). Beyond the centralized mindset. *Journal of the Learning Sciences, 5,* 1–22.

Resnick, M. (1999). Decentralized modeling and decentralized thinking. In W. Feurzeig & N. Roberts (Eds.), *Modeling and simulation in science and mathematics education* (pp. 114–137). New York: Springer-Verlag.

Richey, R. C., & Nelson, W. A. (1996). Developmental research. In D. Jonassen (Ed.), *Handbook of research for educational communications and technology* (pp. 1213–1245). Washington, DC: Association for Educational Communications and Technology.

Richmond, B., & Peterson, S. (1996). *STELLA: An introduction to systems thinking*. Hanover, NJ: High Performance Systems.

Rieber, L. P. (1987). LOGO and its promise: A research report. *Educational Technology, 27*(2), 12–16.

Rieber, L. P. (1990). Using computer animated graphics in science instruction with children. *Journal of Educational Psychology, 82,* 135–140.

Rieber, L. P. (1991). Animation, incidental learning, and continuing motivation. *Journal of Educational Psychology, 83,* 318–328.

Rieber, L. P. (1992). Computer-based microworlds: A bridge between constructivism and direct instruction. *Educational Technology Research & Development, 40*(1), 93–106.

Rieber, L. P. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research & Development, 44*(2), 43–58.

Rieber, L. P., & Parmley, M. W. (1995). To teach or not to teach? Comparing the use of computer-based simulations in deductive versus inductive approaches to learning with adults in science. *Journal of Educational Computing Research, 13*(4), 359–374.

Rieber, L. P., Luke, N., & Smith, J. (1998). Project KID DESIGNER: Constructivism at work through play. *Meridian: Middle School*

Computer Technology Journal [online], *1*(1). http://www.ncsu.edu/meridian/archive_of_meridian/jan98/index.html.

Roschelle, J. (1991, April). *MicroAnalysis of qualitative physics: Opening the black box*. Paper presented at the annual meeting of the American Educational Research Association, Chicago. (ERIC Document ED 338 490)

Roschelle, J., Kaput, J., & Stroup, W. (2000). SimCalc: Accelerating student engagement with the mathematics of change. In M. J. Jacobson & R. B. Kozma (Eds.), *Learning the sciences of the 21st century: Research, design, and implementing advanced technology learning environments* (pp. 47–75). Mahwah, NJ: Lawrence Erlbaum Associates.

Saettler, L. P. (1990). *The evolution of American educational technology*. Englewood, CO: Libraries Unlimited.

Salomon, G., Perkins, D. N., & Globerson, T. (1991). Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Researcher, 20*(3), 2–9.

Spitulnik, M. W., Krajcik, J. S., & Soloway, E. (1999). Construction of models to promote scientific understanding. In W. Feurzeig & N. Roberts (Eds.), *Modeling and simulation in science and mathematics education* (pp. 70–94). New York: Springer-Verlag.

Suppes, P. (1980). Computer-based mathematics instruction. In R. Taylor (Ed.), *The computer in the school: Tutor, tool, tutee* (pp. 215–230). New York: Teachers College Press.

Tetenbaum, T., & Mulkeen, T. (1984, November). Logo and the teaching of problem-solving: A call for a moratorium. *Educational Technology,* 16–19.

Tinker, R. F., & Thornton, R. K. (1992). Constructing student knowledge in science. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 153–170). New York: Springer-Verlag.

van den Akker, J. (1999). Principles and methods of development research. In J. van den Akker, R. M. Branch, K. Gustafson, N. Nieveen, & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 1–14). Dordrecht, The Netherlands: Kluwer Academic.

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

White, B. Y. (1992). A microworld-based approach to science education. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 227–242). New York: Springer-Verlag.

White, B. Y. (1993). ThinkerTools: Causal models, conceptual change, and science education. *Cognition and Instruction, 10*(1), 1–100.

White, B. Y., & Frederiksen, J. R. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction, 16*(1), 3–118.

White, B. Y., & Frederiksen, J. R. (2000a). Technological tools and instructional approaches for making scientific inquiry accessible to all. In M. J. Jacobson & R. B. Kozma (Eds.), *Learning the sciences of the 21st century: Research, design, and implementing advanced technology learning environments* (pp. 321–359). Mahwah, NJ: Lawrence Erlbaum Associates.

White, B. Y., & Frederiksen, J. R. (2000b). Technological tools and instructional approaches for making scientific inquiry accessible to all. In M. J. Jacobson & R. B. Kozma (Eds.), *Innovations in science and mathematics education: Advanced designs for technologies of learning* (pp. 321–359). Mahwah, NJ: Lawrence Erlbaum Associates.

White, B. Y., & Horowitz, P. (1987). *ThinkerTools: Enabling children to understand physical laws*. Cambridge, MA: Bolt, Beranek, and Newman.

Wilensky, U., & Stroup, W. (2002, April). *Participatory simulations: Envisioning the networked classroom as a way to support systems learning for all.* Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.