19. INTELLIGENT TUTORING SYSTEMS: PAST, PRESENT, AND FUTURE

Valerie J. Shute
ARMSTRONG LABORATORY
BROOKS AIR FORCE BASE, TEXAS

Joseph Psotka
U.S. ARMY RESEARCH INSTITUTE
ALEXANDRIA, VIRGINIA

19.1 INTRODUCTION

Many aspects of intelligent tutoring systems (ITS) are addressed in a search for answers to the following main questions: (a) What are the precursors of ITS? (b) What does the term mean? (c) What are some important milestones and issues across the 20+ years of ITS history? (d) What is the status of ITS evaluations? (e) What is the future of ITS? Let's start with an historical perspective.

19.2 PRECURSORS OF ITS

19.2.1 Early Mechanical Systems

Charles Babbage (early 1800s) is typically credited with being the first to envision a multipurpose computer. He dreamed of creating an all-purpose machine, which he called the *analytic engine*. However, because of the technological constraints of the time, he was never able to build his dream, although he did succeed in building a difference engine, an automatic (mechanical) means of calculating logarithm tables.

The notion of using "intelligent machines" for teaching purposes can be traced back to 1926 when Pressey built an instructional machine teeming with multiple-choice questions and answers submitted by the teacher (see 2.3.4.2). It delivered questions, then provided immediate feedback to each learner:

> The somewhat astounding way in which the functioning of the apparatus seems to fit in with the so-called "laws of learning" deserves mention in this connection. The "law of recency" operates to establish the correct answer in the mind of the subject, since it is always the *last* answer which is the right one. The "law of frequency" also cooperates; by chance the right response tends to be made most often, since it is the *only* response by which the subject can go on to the next question. Further, with the addition of a simple attachment, the apparatus will present the subject with a piece of candy or other reward upon his making any given score for which the experimenter may have set the device; that is the "law of effect" also can be made, automatically (see 2.2.1.3), to aid in the establishing of the right answer (Pressey, 1926, p. 375).

While the above system was definitely clever for its time, it could not be construed as intelligent as it was mechanically set with prespecified questions and answers. So, although it was inflexible, this system did incorporate contemporary learning theories and pedagogical strategies into its design (e.g., giving out candy for correct responses).

General-purpose digital computers arose in the mid-1900s, paving the way for truly (artificially) intelligent machines. Basically, these computers consisted of a numerical central processor whose mechanism was electronic, not mechanical, and based on a binary, not decimal, system. They were also characterized by having a built-in ability to make logical decisions, and a built-in device for easy storage and manipulation of data.

During this period of computer infancy, Alan Turing (1912–1954, British mathematician and logician) provided a major link between these modern, digital computing systems and thinking. He described a computing system capable of not only "number crunching" but symbolic manipulation as well. He also developed what is now known as the "Turing test," a means of determining a machine's intelligence. The test consists of an individual asking questions, in real-time, of both a human being and a computer. The interrogator attempts, in any way possible, to figure out which is which via conversations over the communication links. The Turing test has particular relevance to intelligent tutoring systems. The core concept behind the test is whether a reasonable person can distinguish between a computer and a person based solely on their respective responses to whatever questions or statements the interrogator renders. Thus, for a computer to pass the test, it would need to communicate like a human being, which is a nontrivial goal. This line of inquiry has challenged and occupied researchers for more than 20 years, and continues to play a prominent role in the development of ITS (see Merrill, Reiser, Ranney & Trafton, 1992). Other communication-related research includes devising knowledge structuring and hypertext techniques within ITS to provide answers to the many possible questions that students could pose to the system. So, the success of this ITS enterprise really can be measured in a way that is similar to the Turing test: How well can the ITS communicate? We should point out, however, that the goal of ITS is to communicate its embedded knowledge effectively, not necessarily in an *identical* manner as human teachers. In fact, some teachers have great difficulty achieving the effective communication goal themselves.

Concurrent with the gradual emergence of computers on the scene (circa 1950s), educational psychologists began reporting in the literature that carefully designed, individualized tutoring produces the best learning for the most people (e.g., Bloom, 1956; Carroll, 1963; Crowder, 1959; Glaser, 1976; Skinner, 1957; see 35.2). Thus, it was quite a natural development to apply computers to the task of individualized teaching. From the 1970s to the present, ITS has been heralded as the most promising approach to delivering such individualized instruction (e.g., Burton & Brown, 1982; Lewis, McArthur, Stasz & Zmuidzinas, 1990; Shute & Regian, 1990; Sleeman & Brown, 1982; Wenger, 1987; Woolf, 1988; Yazdani & Lawler, 1986). We'll now review what led to the development of "intelligent" computerized instruction.

19.2.2 Programmed Instruction and Computer-Assisted Instruction

In the early 1960s, programmed instruction (PI) was educationally fashionable (see 2.3.4, 22.4.1). This kind of pedagogy related to any structured, goal-oriented instruction. According to Bunderson (1970), PI required the program designer to specify input and output in terms of entering skills and terminal behaviors of the learner. In performing a task analysis, the designer determined the subproblems or component behaviors, as well as their relationships. As learners were led through the problems in the curriculum (lockstep), overt responses were obtained at every step. Incorrect responses were immediately corrected, and learners were always informed of their solution accuracy before moving on to some other content area. Most supporters of the PI technology strongly believed that it would enhance learning, particularly for low-aptitude individuals. However, evidence supporting this belief was underwhelming (see Cronbach & Snow, 1981).

In general, PI refers to any instructional methodology that utilizes a systematic approach to problem decomposition and teaching (e.g., Briggs, Campeau, Gagné & May, 1967; Gagné, 1965; see 2.3.4, 22.4.1). In some cases, PI was embedded in a computer program, known as *computer-assisted instruction* (CAI or computer-based training, CBT). Some similarities between PI and CAI are that both have well-defined curricula and branching routines (intrinsic branching for PI, conditional branching for CAI). A major distinction between the two is that CAI is administered on a computer (see 12.1).

Computer-assisted instruction also evolved from Skinnerian stimulus-response psychology: "... the student's response serves primarily as a means of determining whether the communication process has been effective and at the same time allows appropriate corrective action to be taken" (Crowder, 1959). In other words, at every point in the curriculum, the computer program evaluates whether the student's answer is right or wrong and then moves the student to the proper path. Built-in remediation loops tutor students who are attempting to answer a question incorrectly. If learners answer correctly, they are moved ahead in the curriculum. Figure 19-1 illustrates a typical flow of events in CAI.

The teacher constructs all branching in the program, ahead of time. The normal CAI procedure presents some material to be learned, followed by a problem to be solved that represents a subset of the curriculum. Problem solution tests the learner's acquisition of the knowledge or skill being instructed at that time. The student's answer is compared to the correct answer, then the computer gives appropriate feedback. If the answer is correct, a new problem is selected and presented. If the student answers incorrectly, remediation is invoked that reviews the earlier material, presents simpler problems that graduate to the depth of the original material, and so forth. Remediation usually requires some attempt to find the source of the error and to treat it specially.

As can be seen in Figure 19-1, there are several places where this simple model may be expanded to create more flexibility and, hence, render it adaptive to individual learners. For instance, various mastery criteria can be imposed, where subjects have to answer a certain proportion of items correctly before moving on. Failure to reach a criterion would force the student back into remediation mode (see "If incorrect" branch) where a different problem is presented, rather than the problem that caused the error.

19.2.3 Intelligent Computer-Assisted Instruction

To distinguish between simple versus more adaptive CAI (i.e., "intelligent" computer-assisted instruction, ICAI), Wenger (1987) pointed out that actually there is no explicit demarcation between the two. Instead, there's a continuum, from linear CAI to more complex branching CAI, to elementary ICAI, to autonomous (or stand-alone) ICAI. This continuum is often misconstrued as representing a worse-to-better progression. Yet, for some learning situations and for some curricula, using fancy programming techniques may be like using a shotgun to kill a fly. If a drill-and-practice environment is all that is required to attain a particular instructional goal, then that's what should be used.

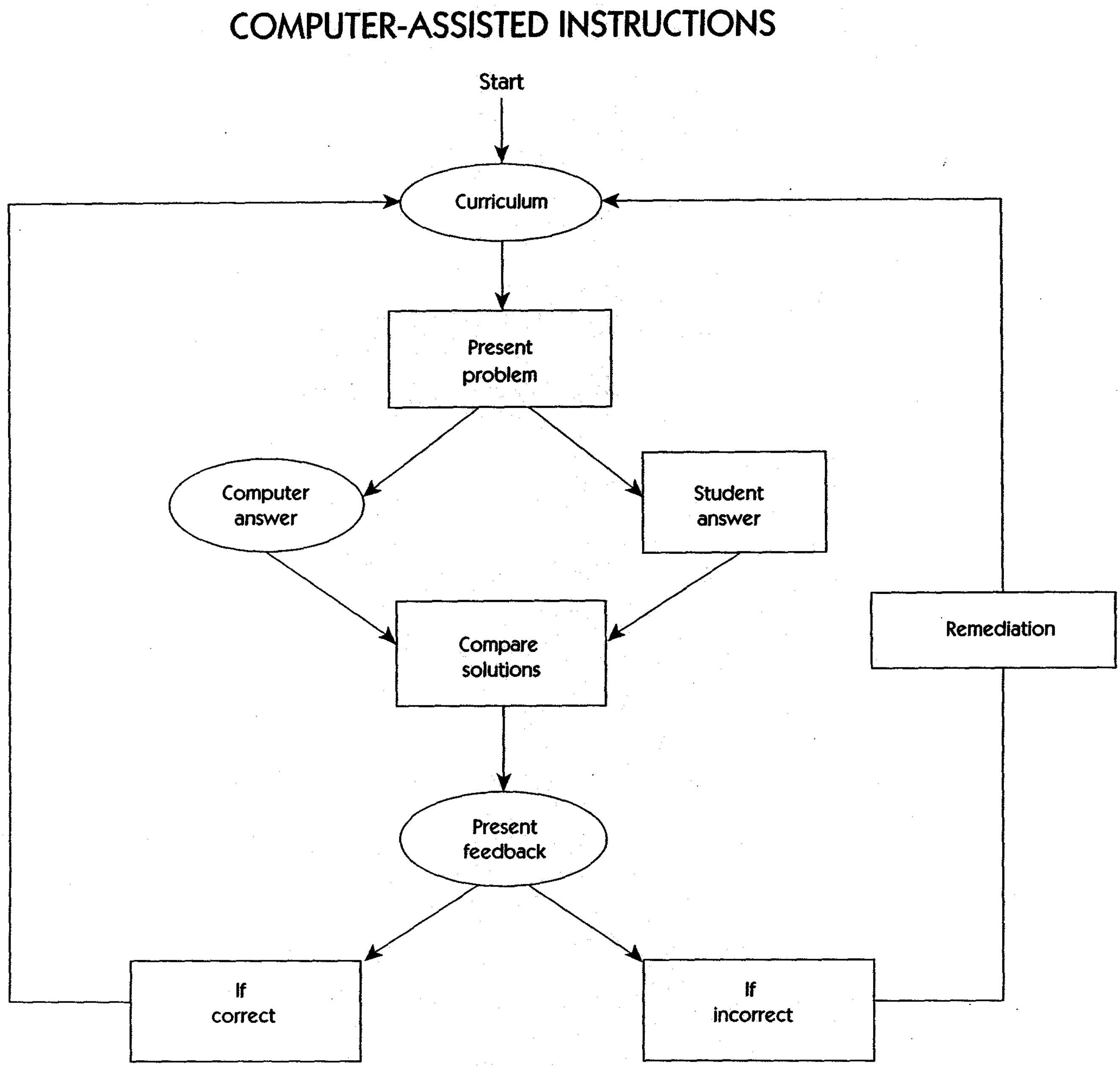

Figure 19-1. Computer-assisted instructions: boxes = program actions; ellipses = canned program knowledge.

Suppose you wanted to build a computerized instructional system to help second-graders learn double-digit addition. If student A answered the following two problems as: 22 + 39 = 61, and 46 + 37 = 83, you'd surmise (with a fair amount of confidence) that A understood, and could successfully apply, the "carrying procedure." But consider some other responses. Student B answers the same problems with 51 and 73; student C answers with 161 and 203; and student D answers with 61 and 85. Simple CAI systems may be incapable of differentiating these incorrect solutions, and remediation would require all three students to redo the specific unit of instruction. But a big problem with this approach is that, typically, there is little difference between the remedial and original instruction. That means that a student who didn't get it right the first time may not get it right the next time if the same instruction and similar problems are used.

A more sensitive (or *intelligent*) response by the system would be to diagnose/classify B's answer as a failure to carry a 1 to the tens column, C's answer as the incorrect adding of the ones column result (11 and 13) to the tens column, and D's as a probable computational error in the second problem (mistakenly adding 6 + 7 = 15 instead of 13). An intelligent system would remediate by specifically addressing each of the three qualitatively different errors.

19.2.3.1. Artificial Intelligence and Cognitive Psychology. How can a computer system be programmed to perform intelligently? This question drives the empirical and engineering research in a field called *artificial intelligence* (AI). The simplest definition is that, "Artificial intelligence is the study of mental faculties through the use of computational models" (Charniak & McDermott, 1985, p. 6). One of the main objectives of AI is to design and develop computer systems that can solve the same kinds of activities that we deem intelligent (e.g., solving a math problem like the one illustrated above, understanding natural language, programming a computer to perform some function(s), maneuvering an aircraft through obstacles, planning a wedding reception, and so forth; see also 22.4.4). There are far too many AI applications to delineate in this chapter. For our purposes, AI techniques relevant to ITS include those dealing with the efficient representation, storage, and retrieval of knowledge (i.e., a large collection of facts and skills—correct and buggy versions), as well as the effective communication of that information. In addition, AI techniques can include inductive and deductive reasoning processes that allow a system to access its own database to derive novel (i.e., not programmed) answers to learners' queries.

Cognitive psychology also provides part of the answer to the question of how to get a computer to behave intelligently by examining issues related to the representation and organization of knowledge types in human memory. Research in this area provides detailed structural specifications for implementation in intelligent computer programs. Cognitive psychology also addresses the nature of errors, a critical feature in the design of intelligent systems to assist learners during the learning process (see 12.2.3, 32.5.3).

19.2.3.2. The Nature of Errors. The idea that students and trainees make mistakes that have to be corrected is fundamental to teaching and learning. Something so fundamental ought to be strongly resistant to change, so it is really quite surprising how the idea of a mistake or error has undergone radical change over the past 2 decades of ITS development. The traditional view of errors encompassed many kinds—from inexplicable accidents, to deliberate inaccuracies—but the most widely held view was that remedial errors stemmed from inaccurate or insufficient knowledge. Remediation then corrected the mistake by providing the correct knowledge or overriding the inaccuracy. The first major shift that occurred in this view began with the development of a theoretical position that errors arose because of complex organizations in knowledge structures that were not wrong, in the traditional sense, but represented the best a student could have at that stage of cognitive development. These developmentally appropriate knowledge structures were called *misconceptions,* and they were soon analyzed in a broad range of sciences (e.g., Aristotelian versus Newtonian physics, studies of heat and temperature) and practical training environments (automobile repair, radar maintenance).

This view of error was explicated in great detail in a series of analyses and experiments by Barbara White and John Frederiksen (1987) in their QUEST system for analyzing levels of understanding of electrical functioning into graduated mental models. Their analyses were actually implemented as qualitative models of the electrical activity in automobile ignition circuits. Simple models, or models that occur developmentally early in the growth of knowledge, were not only incomplete; they were also wrong or inconsistent in basic ways. They could not easily be transformed into more complete models. Yet, the simple models effectively captured the knowledge of novices as they moved on the road to expertise, so it is not clear if these models could have been improved at that stage of development. Thus, it appeared that error or inconsistency was necessary in the growth of knowledge.

As they demonstrated, it took a great deal of effort to conduct error analysis with sufficient scope and detail to be able to arrive at such complete models. It is perhaps for this reason that no other example comes close to duplicating their feat. Yet, the intellectual implications of graduated mental models as the basis for misconceptions and error is stunningly apparent for whoever next decides to pick up the challenge and analyze knowledge structures into such progressive systems.

An alternative conception of error that has developed contemporaneously with the misconception literature is that of a *buggy algorithm.* Work in this area began with Burton and Brown's seminal simulation—*How the West Was Won*—where certain strategic and algorithmic bugs were identified in student play. A specific program was written, DEBUGGY, that attempted to identify and remediate these bugs (Brown & Burton, 1978; Burton, 1982). Unlike the work on misconceptions and graduated mental models, bugs were simpler deconstructions in smaller semantic networks of skills.

This analysis of errors has had a productive life of its own in the work of Soloway (catalogs of bugs, Johnson & Soloway, 1984), Sleeman (mal-rules, Sleeman, 1987), and VanLehn (impasses, VanLehn, 1990). It continues strongly in the model-tracing technology of John Anderson's various tutors (e.g., Anderson, 1993) where bug catalogs or lists of errors are embedded in specific production system rules that manage all interactions between the student and tutor. Anderson has proclaimed a much broader view to encompass not only errors but also all cognitive skills. His position is, simply stated, that cognitive skills are realized by production rules. Not only errors, but all skills as well, are decomposable into unitary rules that fit into a grand cognitive architecture dominated by production rules.

VanLehn's work on *impasses* extends this buggy conception of errors by analyzing the ways these errors are generated (VanLehn, 1990). Oversimplifying his analysis somewhat, VanLehn's framework can be described by saying that bugs are the result of unsuccessful attempts to extend existing rules to apply to novel situations (repairs). These repairs can be modeled and predicted by impasse theory to predict students bugs and problem solving. Usually the repairs are simple actions, such as removing an action step in the production rules, substituting an operator, or deleting a variable.

The final view of errors that has evolved along with ITS sees the error as a result of insufficient support given to the student. When a student learns a new skill or body of knowledge, it is through the support of teachers, students, or other parts of the environment. This environment acts as a general scaffolding to strengthen the student's first new skills or knowledge structures (Palincsar & Brown, 1984). It also provides the context that makes the skills or knowledge meaningful. Some of this scaffolding lies literally in the minds of the other students or teachers, or more precisely, between the minds of everyone. As a kind of social group think, the ideas and scaffolding are part of the total situation (Brown, Collins & Duguid, 1989) and so it has been called *situated cognition.* If the environment is literally part of the skills and knowledge, then changing it abruptly can actually change student thinking and lead directly to errors.

This fascinating research related to different kinds of errors owes its existence directly to the practical and theoretical developments that ITS has spawned. All have real import for the design of instruction, but at the moment, they are still very distant from each other and show no real signs of converging into a common theoretical framework.

19.2.3.3. Summary. Branching is a fundamental aspect of PI, CAI, and ICAI. It recognizes the fact that knowledge is interrelated in many complex ways, and there may be multiple good paths through the curriculum. AI programming techniques empower the computer to manifest intelligence by going beyond what's explicitly programmed, understanding student inputs, and generating rational responses based on reasoning from the inputs and the system's own database.

In the example just provided, prior to teaching double-digit addition, the system could first ascertain if the learner was skilled (to the point of automaticity) with single-digit addition, drilling the learner across a variety of problems, and noting accuracy and latency for each solution. Subsequently, it may be effective to introduce: (a) double-digit addition without the carrying procedure (23 + 41), (b) single- to double-digit addition (5 + 32), or (c) single-digit addition to 10 (7 + 10). Each of these curriculum elements is warranted, and some are easier to grasp than others. However, for more complex knowledge domains, such as history, or the scientific debate over the extinction of dinosaurs, the complexity of alternatives is beyond enumeration. And it is the complexity of this branching that really provides a qualitative break between older forms of PI and CAI and newer ITS. Not only is the branching in ITS complex, it is also algorithmic and not enumerated, predefined, or hand crafted. With this qualitative increase in complexity comes a flexibility of interaction and potential for communication that, better than anything else before, begins to qualify for the word *intelligent.*

Another aspect of computer intelligence deals with the identification and remediation of errors (bugs) in a learner's knowledge structure or performance. The simple illustration with four hypothetical students shows the possible power of adding AI to instructional software that can recognize bugs or misconceptions via: (a) a bug catalog that specifically recognizes each mistake (e.g., Johnson & Soloway, 1984), (b) a set of mal-rules that define the kinds of mistakes possible with this set of problems (e.g., Sleeman, 1987), or (c) a set of production rules that specifically anticipate all alternative problem solutions and can respond to each one (e.g., Anderson, 1993; VanLehn, 1990). Each of these will be discussed in more detail in the section of this chapter outlining the 20-plus years of history of ITS. First, we need to operationalize some terms.

19.3 INTELLIGENT TUTORING SYSTEMS DEFINED

While many researchers in the field view ICAI and ITS as interchangeable designations, we make a subtle distinction between the two: ITS represents a more specific type of ICAI, due to the attributes discussed below.

19.3.1 Early Specifications of ITS

An early outline of ITS requirements was presented by Hartley and Sleeman (1973). They argued that ITS must possess: (a) knowledge of the domain (expert model), (b) knowledge of the learner (student model), and (c) knowledge of teaching strategies (tutor). It is interesting to note that this simple list has not changed in more than 20 years (see Lajoie & Derry, 1993; Polson & Richardson, 1988; Psotka, Massey & Mutter, 1988; Regian & Shute, 1992; Sleeman & Brown, 1982).

All of this computer-resident knowledge marks a radical shift from earlier "knowledge-free" CAI routines. Furthermore, the ability to diagnose errors and tailor remediation based on the diagnosis represents a key difference between ICAI and CAI. Figure 19-2 illustrates these knowledge components and their relations within a generic ITS. Each of these ITS components will be discussed, in turn.

19.3.2 ITS Components and Relationships

A student learns from an ITS primarily by solving problems—ones that are appropriately selected or tailormade—that serve as good learning experiences for that student. The system starts by assessing what the student already knows, the *student model.* The system concurrently must consider what the student needs to know, the *curriculum* (also known as the *domain expert*). Finally, the system must decide which curriculum element (unit of instruction) ought to be instructed next and how it shall be presented, the *tutor* or inherent teaching strategy. From all of these considerations, the system selects, or generates, a problem, then either works out a solution to the problem (via the domain expert) or retrieves a prepared solution. The ITS then compares its solution, in real time, to the one the student has prepared and performs a diagnosis based on differences between the two.

Feedback is offered by the ITS based on such student-advisor considerations as how long it's been since feedback

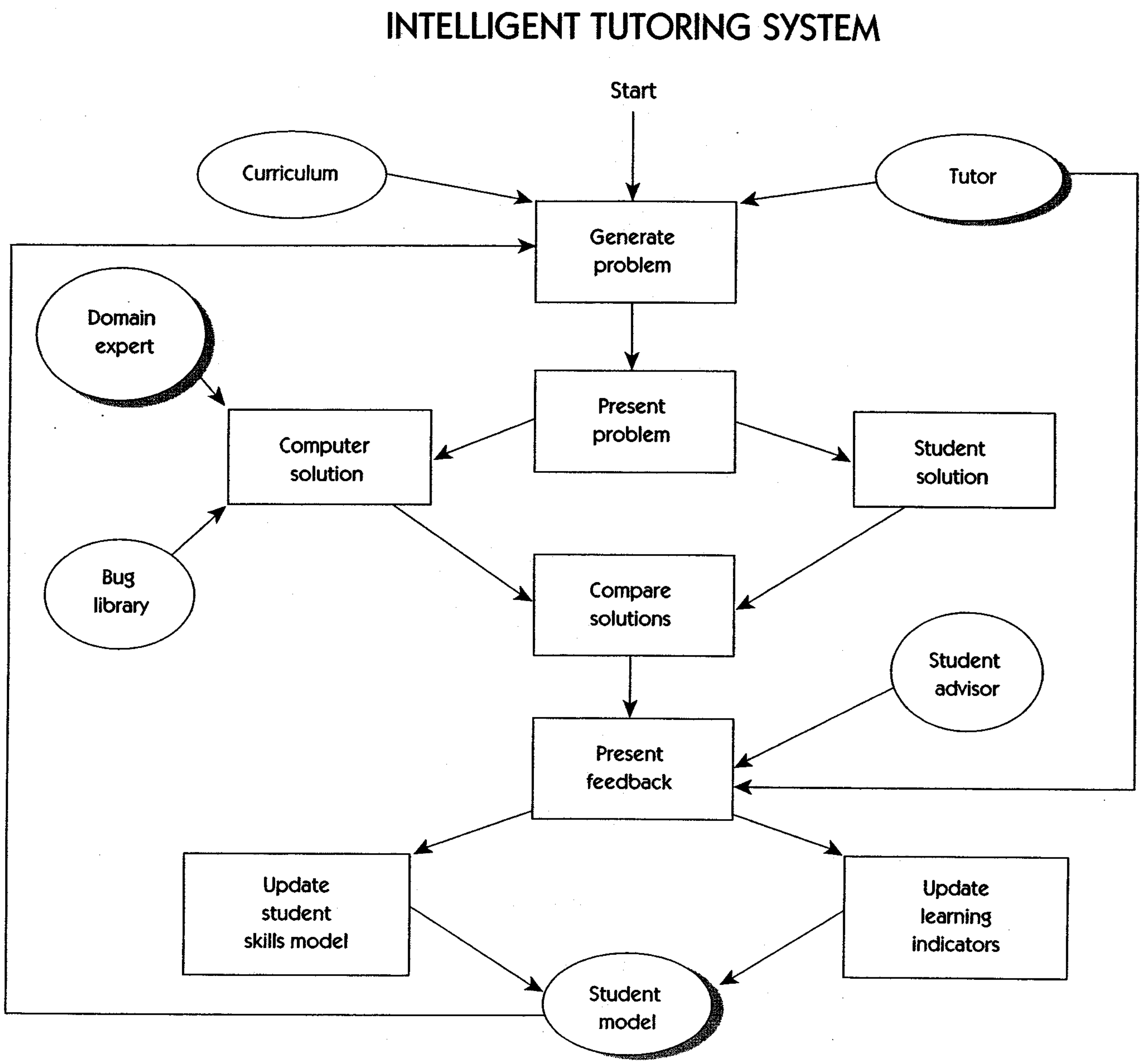

Figure 19-2. Intelligent tutoring system: boxes = program decisions & actions; ellipses = program knowledge bases; shaded ellipses = core ITS components.

was last provided, whether the student already received some particular advice, and so on. After the feedback loop, the program updates the student skills model (a record of what the student knows and doesn't know) and increments learning progress indicators. These updating activities modify the student model, and the entire cycle is repeated, starting with selecting or generating a new problem (see 32.3).

Not all ITS include these components, and the problem-test-feedback cycle does not adequately characterize all systems. However, this generic depiction does describe many current ITS. Alternative implementations exist, representing conceptual as well as practical differences in their design. For example, the standard approach to building a student model involves representing emerging learner knowledge and skills. The computer responds to updated observations with a modified curriculum that is minutely adjusted. Instruction, therefore, is very much dependent on individual response histories. But an alternative approach involves assessing *incoming* knowledge and skills, either instead of, or in addition to, emerging knowledge and skills. This alternative enables the curriculum to adapt to both persistent and/or momentary performance information as well as their interaction (see Shute, 1993a, 1993b, 1995). In fact, many have argued that incoming knowledge is the single most important determinant of subsequent learning (e.g., Alexander & Judy, 1988; Dochy, 1992; Glaser, 1984).

Other kinds of systems may not even have a tutor/coach present. For example, the strength of microworlds (exploratory environments) resides in the underlying simulation and explicit interfaces in which students can freely conduct experiments and obtain results quickly and safely (see 12.3). This is a particularly attractive feature for domains that are hazardous, or do not frequently occur in the real world. Furthermore, these systems can be intrinsically motivating, in terms of generating interesting complexities that keep students interested in continuing to explore, while giving them sufficient success to prevent frustration.

19.3.3 The "I" in ITS

Our working definition of computer-tutor intelligence is that the system must *behave* intelligently, not actually *be* intelligent, like a human being. More specifically, we believe that an intelligent system must be able to: (a) accurately diagnose students' knowledge structures, skills, and/or styles using principles, rather than preprogrammed responses, to decide what to do next; and then (b) adapt instruction accordingly (e.g., Clancey, 1986; Shute, 1992; Sleeman & Brown, 1982). Moreover, the traditional intelligent tutoring system ". . . takes a longitudinal, rather than cross-sectional, perspective, focusing on the fluctuating cognitive needs of a single learner over time, rather than on stable inter-individual differences" (Ohlsson, 1986, pp. 293–294).

In order to obtain a rough idea of the degree of consensus among researchers in the ITS community, 20 experts were asked to summarize, in a couple of sentences, their ideas on what the "I" in ITS meant. Following are the different responses received (in alphabetical order, and slightly edited, for readability).

Ton de Jong: "Intelligent" in ITS stands for the ability to use (in a connected way) different levels of abstraction in the representation of the learner, the domain, and the instruction. The higher the range of abstraction, the higher the intelligence. The phrase "in a connected way" implies that one should be able to go from specific (e.g., log files) to abstract (e.g., learner characteristics), as well as the other way around (e.g., from general instructional strategies to a specific instructional transaction).

Sharon Derry: An intelligent instructional system can observe what the student is doing during problem solving and/or has done over a series of problem-solving sessions, and from this information draw inferences about the student's knowledge, beliefs, and attitudes in terms of some theory of cognition. A system can be intelligent whether or not it makes instructional decisions based on this information, but if it *doesn't* use such information in instructional decision making, then I don't think of it as a tutoring system, but rather a tool that has some diagnostic capabilities.

Wayne Gray: I concede a wide latitude on the application of the term "ITS" in regard to instructional systems. However, at some level and to some degree, there should be some sort of "cognitive modeling" technology involved. The modeling can be of an ideal student, instructor, or grader, or of a less-than-ideal problem solver as in the "student models" that are often built up in ITS. To be intelligent, a system has to incorporate and use a model for making decisions about what to do at any given point during learning.

Lee Gugerty: Intelligent tutoring involves: (a) explicit modeling of expert representations and cognitive processes; (b) detection of student errors; (c) diagnosis of students' knowledge (correct, incorrect, and missing); (d) instruction adapted to students' knowledge state (via problem selection, hints, feedback, and explicit didactic instruction); and (e) doing all of the above in a timely fashion as the student solves problems (not post hoc).

Pat Kyllonen: An "intelligent" tutoring system is one that uses AI programming techniques or principles. However, what is considered AI (as opposed to standard) programming changes over time (e.g., expert systems used to be archetypal AI systems but are now found in $100 PC software packages). For me, two features separate ITS software from conventional CAI. One is the existence of a student model. What the student knows cannot be recorded directly but must be inferred by the system, based on a pattern of successes or failures by the student and an "understanding" of what knowledge problems in the curriculum call upon. Another feature is the existence of "coaches," "demons," or "bug libraries" that can observe a student's behavior and either diagnose the behavior in terms of the student's current knowledge structure or suggest corrections to that behavior.

Susanne Lajoie: The "I" in ITS means that the computer can provide adaptive forms of feedback to the learner based on a dynamic assessment of the student's "model" of performance. Intelligent feedback means that the assessment of the learner is ongoing, the feedback is appropriate to that particular learner in the context of where an impasse has been encountered, and it is not canned but generated on the spot, based on student needs.

Alan Lesgold: "Intelligent" means that the system uses inference mechanisms to provide coaching, explanation, or other information to the student performing a task. Further, it implies that this information is tuned to the context of the student's ongoing work and/or a model of the student's evolving knowledge.

Matt Lewis: An "intelligent" tutoring system contains, at a minimum, a reasonably general simulation of human problem solving in direct service of communicating knowledge and, like a good human tutor, separates domain knowledge from pedagogical knowledge. The simulation might solve domain-specific problems in the target instructional domain (e.g., a humanlike approach and solution to the problem of writing a fugue) or solve pedagogical problems (e.g., error diagnosis and attribution, or selection of appropriate response).

Wes Regian: An ITS differs from CAI in that: (a) instructional interactions are individually tuned at run time to be as efficient as possible, (b) instruction is based on cognitive principles, and (c) at least some of the feedback is generated at run time, rather than being all canned. It is not particularly important to me what language the system is written in, whether or not the system is in any sense arguably aware of anything, and whether its decisions are rendered in a manner that is the same as a human decision.

Frank Ritter: The "I" in ITS usually indicates that a single knowledge-based component has been added that helps a tutoring system perform one aspect of its performance in a better way. This can be in lesson scheduling, providing examples of domain knowledge in action, or providing domain knowledge for comparison with a student's behavior. What it *should* mean is that it does the whole job intelligently. The systems are usually not systems in the full sense of the word; they tend to be prototypes, with whole parts missing.

Derek Sleeman: "Intelligent" tutoring systems need to have motivating learning environments, to communicate effectively, and to render dynamic decisions about appropriate control strategies. Since the 1960s, we've seen that the same material delivered on various systems differentially invoke motivation; thus we need to confirm the factors that impact a learner's motivation. Next, communication can only occur when there's a shared world view. In conventional dialogues, human beings dynamically tailor their language to the person to whom they are speaking, but computers are not yet so adaptable. Finally, control implies which of the partners in the dialogue will take the initiative, and it's often necessary to change control during an interaction, depending on the social setting, the student's motivation, and the level of incoming knowledge.

Elliot Soloway: The intent of the "I" in ITS was to explicitly recognize that a tutoring system needs to be exceedingly flexible in order to respond to the immense variety of learner responses. CAI, as the forerunner of ITS, didn't have the range of interactivity needed for learning. In fact, the movement from ICAI to ITS was to further distance the new type of learning environments from the rigidity of CAI.

Sig Tobias: "Intelligent," in an ITS context, means that the program is flexible in the method and sequence with which instructional materials are presented to the student. Furthermore, the system is capable of adapting instructional parameters to student characteristics by using data collected prior to, or during, instruction for such decisions. Finally, it suggests that the instructional system can advise the student regarding options most likely to be successful for the student.

Kurt VanLehn: "Intelligent" means that at least one of the three classic modules is included in the tutoring system. That is, the machine has either a subject-matter expert, a diagnostician/student modeler, or an expert teacher. Just as in any AI system, an expert system with only 10 production rules is intelligent only in that it holds the possibilities for expansion; a 100-rule system is moderately intelligent; and 1,000+ rules means you're really getting there.

Beverly Woolf: My view of tutor intelligence includes the following elements: (a) mechanisms that model the thinking processes of domain experts, tutors, and students; (b) environments that supply world-class laboratories within which students can build and test their own reality; and (c) a computer partner that facilitates the aha! experience, recognizes the student's intention, and aids and advises the student. An intelligent environment would also support complex discoveries.

As seen in this nonrandom sample of responses about what constitutes intelligence in an ITS, just about everyone agrees that the most critical element is real-time *cognitive diagnosis* (or student modeling). The next most frequently cited feature is *adaptive remediation* (see 22.5). And while some maintain that remediation actually comprises the "T" in intelligent tutoring systems, our position is that the two components (diagnosis and remediation), working in concert, make up the intelligence in an ITS (see our working definition, above). Consider the case where a system diagnoses a student's skill level, but makes no effort to rectify any faulty behaviors. Can that system really be classified as intelligent? Theoretically, perhaps, but practically, no. Other characteristics of intelligence appear less frequently in these responses (e.g., canned vs. generated problems and feedback, degree of learner control in the environment, and presence of awareness).

The degree of agreement among responders was actually surprising given the diversity of respective research interests and backgrounds (computer scientists, psychologists, educators). But this degree of consensus was not always there. Until fairly recently, the field was not only esoteric but also quite fractionated; people disagreed about the definition of "intelligence" in a computer tutor. To understand the current congruence, we need to jump back briefly in time to see the evolution of intelligent tutoring systems, from the late 1960s to the present (mid-1990s).

19.4 THE 20-YEAR HISTORY OF ITS

Instead of discussing individual tutoring systems that spanned this period, we present salient characteristics of systems appearing at various points in time, illustrating with exemplar tutors. For excellent discussions of individual intelligent tutoring systems, see the following books: Bierman, Breuker & Sandberg, 1989; Goodyear, 1991; Lajoie & Derry, 1993; Lawler & Yazdani, 1987; Nickerson & Zodhiates, 1988; Polson & Richardson, 1988; Psotka, Massey & Mutter, 1988; Regian & Shute, 1992; Self, 1988; Sleeman & Brown, 1982; and Wenger, 1987. The issues, by decade, that will be discussed can be seen in Table 19-1.

19.4.1 Up through the 1970s: Defining the Issues

Hardware and software have evolved at an astounding rate over the past 20 years. To put things in perspective, consider the 1970s: "Pong" was the rage (i.e., a simple black-and-white computerized table tennis game) and 8K random access memory (RAM) the norm for a PC. Computer-

TABLE 19-1. IMPORTANT ISSUES RELATED TO ITS DEVELOPMENT

1970s	1980s	1990s
Problem Generation	Model Tracing	Learner Control
Simple student modeling	More buggy-based systems	Individual vs. collaborative Learning
Knowledge representation	Case-based reasoning	Situated learning vs. information processing
Socratic tutoring	Discovery worlds	Virtual reality
Skills & strategic knowledge	Progression of mental models	
Reactive learning environments	Simulations	
Buggy library	Natural language processing	
Expert systems & tutors	Authoring systems	
Overlay models/genetic graph		

administered instruction developed before the 1970s was inflexible and didactic because the systems had very limited capabilities (i.e., memory capacity and computational speed) for adaptive diagnosis and feedback. Furthermore, ". . . the only theory available to guide instructional development was behavior theory, which poorly matched the cognitive goals of education" (Lesgold, 1988, p. iii; see 2.2). Over time, researchers in AI and cognitive psychology joined forces, and together provided a basis for a new generation of computer-based teaching programs. Some of the research issues that dominated the 70s are discussed below (see 5.23).

19.4.1.1 Real-Time Problem Generation. The earliest systems to incorporate some now "classic" ITS elements were programs that generated problems and learning tasks, representing a big departure from the canned problems stored in CAI databases (see also 7.5.2). For example, Uhr (1969) developed a computer-based learning system that created, in real time, simple arithmetic problems and vocabulary recall tasks. The next major advance in this area came in the form of computer programs that generated problems that had been tailored to the knowledge and skill level of a particular student, thus providing the foundation for student modeling.

19.4.1.2. Simple Student Modeling. The Basic Instructional Program (BIP) develops procedural skills required in learning the programming language BASIC (Barr, Beard & Atkinson, 1976). It did so by selecting problems based on what the student already knew (past performance), which skills should be taught next, and its analysis of the skills required (problems in the curriculum). Exercises were dynamically and individually selected per person (from a pool of 100 sample problems). Then teaching heuristics were applied to the student model to identify skills to be taught, and exercises were selected that best involved those skills. Selection of appropriate exercises was based on information contained in a network called the Curriculum Information Network (CIN), relating tasks in the curriculum to issues in the domain knowledge. Thus, a programming task in the tutor was represented in terms of its component skill requirements. Based on a task analysis, BIP knew that the component skills needed for solving a particular programming problem included such skills as: initialize numeric variable, use for-next loop with literal as final value, and so forth. Moreover, each task tapped a number of skills.

19.4.1.3. Knowledge Representation. Classic CAI used pages of text to represent knowledge, but with little psychological validity. In contrast, Carbonell's (1970) SCHOLAR program (often credited with being the first true ITS) used a semantic net to represent domain knowledge (South American geography) as well as the student model. Nodes in the network had tags to indicate whether the concept was known to the student. This novel application of semantic network as a general structure of knowledge representation supported mixed-initiative dialogs with students. Not only could the computer ask questions of the student but the student also could, theoretically, ask questions of the computer. One major limitation of this semantic knowledge representation was the difficulty of representing procedural knowledge (see 5.3).

19.4.1.4. Socratic Tutoring. Carbonell's research spawned another line of work concerned with enabling systems to engage in Socratic dialogues, believed to involve the learner more actively in the learning process. Collins (1977) outlined a set of tutorial rules for Socratic tutoring that were incorporated into a system called WHY (Stevens & Collins, 1977). For example, consider the following IF/THEN string: IF the student gives an explanation of one or more factors that are not sufficient, THEN formulate a general rule for asserting that the given factors are sufficient, and ask the student if the rule is true (Collins, 1977, pp. 343–344). Instead of semantic nets, the domain knowledge (rainfall) was stored in a "script hierarchy" containing information about stereotypical sequences of events.

19.4.1.5. Skills and Strategic Knowledge. Another attempt to stimulate thought among students (rather than being passive recipients of information) was the focus of a group of researchers at Xerox PARC in the mid- to late-1970s. For instance, WEST (Burton & Brown, 1976) was developed to help students learn/practice skills involved in the manipulation of arithmetic expressions. The goal was to move around a game board (*How the West Was Won*) and either advance the maximum number of squares, land on and thus "bump" an opponent back some fixed amount of squares, or take a shortcut. Not only was basic arithmetic skill involved but also strategic knowledge was required. The system was attentive to all levels of knowledge and skill, but the "coach" was somewhat unobtrusive, sitting in the background monitoring the student's moves, intervening only when it was clear that the student was floundering. Then the coach would make a few suggestions to enhance student skills. WEST's coaching goals were accomplished by focusing on the strategy used to construct a move (viz.,"issue-based" tutoring).

19.4.1.6. Reactive Learning Environments. In reactive learning environments, the system responds to learners' actions in a variety of ways that extend understanding and help change entrenched belief structures using examples that challenge the learner's current hypotheses. An early, excellent example of this kind of environment was SOPHIE (Sophisticated Instructional Environment), designed to assist learners in developing electronic troubleshooting skills (see Brown & Burton, 1975; Brown, Burton & deKleer, 1982). For instance, in SOPHIE I, learners located faults in a broken piece of equipment. They could ask SOPHIE questions in English (e.g., to obtain values of various measurements taken on the device). SOPHIE I included three main components: a mathematical simulation, a program to understand a subset of natural language, and routines to set up contexts, keep history lists, and so on. A student, troubleshooting a simulated piece of equipment, could offer a hypothesis about what was wrong. SOPHIE I reacted to the request by comparing the hypothesis to the measurements entered by the student. SOPHIE II extended the environment of its predecessor by adding an articulate expert based on a prestored decision tree for troubleshooting the power supply that was annotated with schema for producing explanations. SOPHIE III represented a significant advance; it contained an underlying expert based on a causal model rather than on a mathematical simulation. The importance of this change is that, in SOPHIE I, the simulator worked out a set of equations not using humanlike, causal reasoning, so it wasn't possible for the system to explain its decision in any detail. But SOPHIE III did employ a causal model of circuits to deal with the student feedback deficiency. Research with SOPHIE spawned a lot of later research in troubleshooting, reactive learning environments, and articulate experts.

19.4.1.7. Buggy Library. Brown and Burton (1978) also developed BUGGY, a frequently cited example of a system employing a "buggy" library approach to the diagnosis of student errors. BUGGY was a framework for modeling misconceptions underlying procedural errors in addition and subtraction where students' errors were represented as the results of "bugs" (errors) in an otherwise correct set of procedures. DEBUGGY (Burton, 1982) was developed as an off-line version of the system based on the BUGGY framework's using the pattern of errors from a set of problems to construct an hypothesis concerning a bug, or combination of bugs, from the library that generated the errors. IDEBUGGY (Burton, 1982) was an on-line version of BUGGY, diagnosing the student's procedure bit-by-bit while giving the learner new problems to solve. The major limitation of these kinds of systems was the inability to anticipate all possible misconceptions. Moreover, bugs could appear transient as they were being repaired.

19.4.1.8. Expert Systems and Tutors. MYCIN (Shortliffe, 1976) was a rule-based expert system for diagnosing certain infectious diseases such as meningitis. GUIDON (Clancey, 1979) was constructed to interface with MYCIN for tutoring, interactively presenting the rules in the knowledge base to a student. This tutoring operated as follows: GUIDON described case dialogues of a sick patient to the student in general terms. The student had to adopt the role of a physician and ask for information that might be relevant to the case. GUIDON compared the student's questions to those that MYCIN would have asked, and then responded accordingly.

19.4.1.9. Overlay Models/Genetic Graph. The definition of an overlay model is one of a novice-expert difference model representing missing conceptions. It's typically implemented as either an expert model annotated for missing items, or an expert model with weights assigned to each element in the expert knowledge base. To illustrate how it works, consider WUSOR (Stansfield, Carr & Goldstein, 1976), the name of the on-line coach for the game WUMPUS (Yob, 1975). The WUMPUS player had to traverse through successive caves to locate the hiding Wumpus. Many dangers faced the player (e.g., pits, bats), but the problem could be solved by applying logical and probabilistic reasoning to information obtained along the way. The goal of the game was to shoot an arrow into the Wumpus' hiding spot before you were killed. WUSOR evolved through (at least) three generations, each with a progressively more sophisticated student model. The first version had only an expert and advisor and did not try to diagnose the learner's state of knowledge. The next version (II) incorporated an overlay model (Carr & Goldstein, 1977) where the expertise was represented as rules, and the student's knowledge state was a subset of the expert's knowledge. Goldstein (1979) made the final transformation to WUSOR (III) by including the genetic graph, combining overlay modeling (rule-based representation) with a learner-oriented set of links between curricular elements. *Genetic* related to the notion of knowledge being evolutionary, and *graph* denoted the relationships between parts of knowledge expressed as links in a network. A genetic graph could represent type of links (e.g., generalization, analogy, refinement) as well as deviation links (i.e., buggy rules as opposed to simply absent ones).

The 1970s were marked by experimental systems that bore little resemblance to one another. During the following decade, systems became less idiosyncratic, but there was still a lot of diversity in the field.

19.4.2 1980s: Standardized Approaches and Environments

The 1980s were characterized by enormous growth and momentum in the ITS field. By the mid-1980s, the development of tutors greatly exceeded their evaluations; everyone wanted to participate in the excitement of building ITS, but few cared to test their system's efficacy (Baker, 1990; Littman & Soloway, 1988; see 12.2, 39.4). Sleeman (1984) attempted to focus research efforts by outlining four main problems with ITS at the time:

1. *Feedback specificity.* Instructional feedback was often not sufficiently detailed for a particular learner.
2. *Nonadaptability.* Systems forced students into their own conceptual framework rather than adapting to a particular student's conceptualization.
3. *Atheoretical foundation.* Tutoring strategies used by the systems lacked a theoretical cognitive foundation.
4. *Restrictive environment.* User interaction and exploration was too restricted.

These main criticisms were addressed, to varying degrees, during the 1980s.

19.4.2.1. Model Tracing. Anderson and his colleagues at Carnegie-Mellon University developed a model-tracing approach to tutoring based on production systems as a way of modeling student behavior. The model-tracing approach has been employed in a variety of tutoring systems, such as the LISP tutor (Anderson, Boyle & Reiser, 1985) and the geometry tutor (Anderson, Boyle & Yost, 1985). Model tracing provides a powerful way both to validate cognitive theories (e.g., Anderson, 1987) and to deliver low-level, personalized remediation. The approach works by delineating many hundreds of production rules that model curricular "chunks" of cognitive skill. A learner's acquisition of these chunks is monitored (i.e., the student model is traced), and departure from the optimal route is immediately remediated.

In theory (and practice), the model-tracing approach for the geometry and LISP tutors is so complete that it captures an enormous percentage of all students' errors. A major drawback is that this approach does not allow students to commit those errors themselves. As soon as there is a misstep, the tutor cries "foul" and stops the student from doing anything else until the correct step is taken. As Reiser points out (e.g., Reiser, Ranney, Lovett & Kimberg, 1989), the student is not only prevented from following these mistakes to their logical conclusion (and getting hopelessly confused) but also prevented from obtaining an insight into the mistake (i.e., that the mistake is obvious). These are some of the best learning experiences students can have, but they appear to be blocked by the model-tracing approach.

Model tracing challenges the first criticism (feedback specificity). That is, the grain size of feedback is as small as you can get (i.e., the production level), thus providing the most detailed, specific feedback possible. However, in some cases (i.e., for certain students or particular problems), this level of feedback may be too elemental; the *forest is lost for the trees.* Next, as mentioned above, the systems can adapt to a wide range of student conceptualizations, challenging the second (nonadaptability) criticism. The approach also demolishes the third criticism (atheoretical foundation), as it was explicitly based on Anderson's cognitive theory (ACT*). The positive features of this approach, however, are achieved at the expense of the fourth (restrictive environment) criticism. That is, the model-tracing approach is restrictive. To accomplish the necessary low-level monitoring and remediation of this approach, the learner's freedom has to be curtailed. So, learning by one's mistakes is out (which is often a powerful way to learn). A final drawback of this approach is that, while it works very well in modeling procedural skill acquisition, it does not work well for domains that are ill-structured, or that are not rule-based (e.g., creative writing, economics, Russian history).

19.4.2.2. More Buggy-Based Systems. During this time period, a plethora of tutors was developed based on the "buggy" library approach (see BUGGY, above). While these systems do provide very specific feedback about the nature of the learner's error (countering criticism 1, feedback specificity), the system response is dependent on the program's ability to match the student's error with that of a stored "bug." Along these same lines, as with model tracing (because only stored bugs are acknowledged), novel bugs are ignored; thus there is no way to update the buggy library or adapt to the learner's current conceptualization (criticism 2, nonadaptability). This approach is theoretically based on the notion of cognitive errors in specific procedures, impasse learning, and repair theory (VanLehn, 1990), countering criticism 3 (atheoretical foundation). Finally, these systems constrain the learner somewhat less than the model-tracing approach; thus, it is a response to criticism 4 (restrictive environment).

A good illustration of a system based on the buggy approach is PROUST (Johnson, 1986; Littman & Soloway, 1988), designed to diagnose nonsyntactic student errors in Pascal programs. The system works by locating errors in students' programs where they compute various descriptive statistics such as the minimum and maximum values, and averages. The major drawback of this system is that it is implemented off line. In other words, the tutor has access to a final product on which to base its diagnosis of student errors: Completed student programs are submitted to PROUST, which prints out the diagnosis (Johnson & Soloway, 1984).

A parallel "buggy" research project involved a system called PIXIE (Sleeman, 1987), an on-line ITS based on the Leeds Modeling System (LMS), a diagnostic model for determining sources of error in algebra problem solving due to incorrect procedural rules or "mal-rules." While some may equate mal-rules with buggy rules, they differ in a fundamental way. Sleeman created them by postulating a set of basic buggy rules from which higher-order mal-rules could be generated from the structure of the knowledge base itself. Mal-rules are inferred from basic principles and bugs; they are at a level of abstraction above bugs. In fact, John Anderson makes the same point about his model-tracing procedures. Because of the complexity of his model-tracing productions, many productions fire or are used over and over again in contexts for which they were not first generated, and so they too take on a kind of abstract or general quality in his framework. The major problem with LMS is that it only diagnoses the incorrect rules; it does not remediate.

19.4.2.3. Case-Based Reasoning. Another category of systems emerging at this time came from case-based reasoning (CBR) research (Schank, 1982; Kolodner, 1988). Proponents of this approach suggest that the goal of ITS should be to teach cases and how to index them. Given that the student, not the program, is the one doing the indexing, this system affords the learner greater freedom and promotes a more adaptive learning environment (countering criticisms 4, restrictive environment, and 2, nonadaptability, respectively). Furthermore, whereas the model-tracing tutors work poorly in ill-structured domains, CBR works well in those areas (e.g., politics, philosophy). This tradeoff, however, can result in less specific feedback to learners (criticism 1, feedback specificity).

These CBR systems also perform well in domains where there are too many rules, or too many ways in which rules can be applied (e.g., programming, game playing). CBR suggests *approximate* answers to complex problems, thereby limiting how many rule combinations should be explored. There are two main processes involved with CBR: indexing (labeling new experiences for future retrieval) and adaptation (changing a retrieved case to fit a current situation). Further, two kinds of indices are required: concrete and abstract. Concrete indices refer to objects and actions usually directly mentioned in the case, while abstract indices refer to more general characterizations. The "indexing problem" deals with ways to determine the correct abstract and concrete indices for cases. How one indexes new cases determines what cases one will compare the inputs against. Using a general index, one can retrieve a case even when it shares no specific details with the current situation.

Schank has made some very provocative statements about the human mind as a storyteller, and about the need to encapsulate knowledge into stories, not into hierarchical data structures like semantic networks. But his procedures have yet to lead to any of the other strong characteristics of ITS that we emphasize in this paper: student models, teaching models, bugs, and so on. Instead, they exist as very generative and interesting systems. As such, they have something in common with microworlds; that is, people enjoy exploring them and can learn from them, particularly those regarding ill-structured and complex domains. However, when students don't learn, or manifest some misconception(s), the very same looseness of structure and organization in these systems prevents them from determining why, and doing something about it. Finally, according to Riesbeck and Schank (1990), case-based reasoning (CBR) serves as a *model* of cognition and learning. But, while these systems present a provocative and well-conceived approach that has many practical and obvious merits, they cannot be said to possess a solid theoretical foundation (criticism 3, atheoretical foundation). A major limitation of this approach includes the problem of anticipating and representing a sufficient number of cases to be cataloged.

19.4.2.4. Discovery Worlds. With just a few exceptions, learning from computers in the 1960s and 1970s was characterized by inflexible presentations of didactic material. But an opposition movement arose in the 1970s that gained steam in the 1980s; it resulted in the development of discovery learning environments (see 7.4.1). These computerized systems (typically a computer simulation environment with simple interface and tools) were designed to make it possible for students to acquire various knowledge and skills on their own. For example, students could learn LOGO (Papert, 1980; see 12.3.2, 24.5) or Newton's laws of motion (White, 1984) within discovery (or micro) worlds. Typically, feedback was "natural" or implicit, not specifically explained to the learner (relating to criticism 1, feedback specificity).

One of the main strengths of these systems was their great adaptability to a range of different learners (countering criticism 2, nonadaptability). Students were free to explore and act within the microworld as they chose; with the ramifications of their actions immediately revealed, countering criticism 4 (restrictive environment). This movement was based on the theoretical premise that in discovery learning, one can radically alter the perceptual relationship between the learner and the knowledge or skills to be acquired, thus addressing criticism 3 (atheoretical foundation). This position was epitomized by Piaget (1954), who stated that ". . . an education which is an active discovery of reality is superior to one that consists merely in providing the young with ready-made wills to will with, and ready-made truths to know with."

A major drawback of these systems is that not all persons are skilled in the requisite inquiry behaviors necessary to achieve success in these environments (see Shute & Glaser, 1990). That is, to be successful, an individual should be able to: formulate efficient experiments; state, confirm, and/or negate hypotheses; appropriately relate hypotheses and experiments; plan future experiments and tests; engage in self-monitoring; and so on.

19.4.2.5. Progression of Mental Models. White, Frederiksen, and their colleagues (Frederiksen, White, Collins & Eggan, 1988; White & Frederiksen, 1987; White & Horowitz, 1987) incorporated ideas from: (a) AI research on mental models and (b) qualitative reasoning to develop QUEST (Qualitative Understanding of Electrical System Troubleshooting) as well as "Thinker Tools." This approach, like model tracing, above, is thus theoretically grounded (in opposition to criticism 3, atheoretical foundation).

These systems work by motivating students to want to learn by pointing out errors and inconsistencies in their current beliefs. Then students are guided through a series of microworlds, each more complex than the one preceding, toward the objective of more precise mental models (see 5.3.7) of the evolving subject matter (e.g., electrical concepts or Newtonian mechanics). Finally, students formalize their developing mental models by evaluating a set of laws describing phenomena in the microworld; then they apply the selected law to see how well it predicts real-world events.

These systems promote learning, neither completely free nor overly restricted (relating to criticism 4, restrictive environment), that resides about halfway between true discovery environments and model-tracing environments. A programmed series of mental models produces higher-level feedback compared to, for example, feedback at the production level (addressing criticism 1, feedback specificity). Finally, the systems can adapt to a wide range of learner misconceptions (challenging criticism 2, nonadaptability).

19.4.2.6. Simulations. Graphical simulations have become more central to the ITS enterprise as the power of computers has grown. Along with increasing computational power, software systems have grown more complex; object-oriented systems can now mimic devices of great complexity and interactivity. Simulations are useful wherever real objects are involved in a learning or training task, and they provide many benefits over real devices. Not only are they less dangerous, less messy, and exactly replicable; simulations are inspectable and self-explanatory in ways that real objects cannot be. Simulations not only display aggregate behavior, but they also are decomposable into constituents that mimic novice or expert mental models. This decomposability of graphic displays and simulations mimics the power of productions in expert systems for creating natural chunks that promote learning (see 17.4).

Early ITS, like SOPHIE, could generate only very simple line drawings. A dramatic increase in the power of graphic simulations took place with Steamer (Hollan, Hutchins & Weitzman, 1984) and the use of personal LISP machines. These machines could generate interactive graphics with animated components. It was not long before this graphical power became available for ITS on smaller personal computers that could be used in industrial and educational settings. Of course, more powerful systems that were developed in the 1980s, like Hawk MACH-III, could expand the number of components and complexity of the animations by orders of magnitude (Kurland, Granville & MacLaughlin, 1992). Using object-oriented constructions, MACH-III made each part of complex radar systems inspectable and self-explanatory. For teaching troubleshooting, each decomposable part of the radar device could even explain its role in the troubleshooting sequence for any fault that had been created in the system. Given this power and complexity, these systems were stretched to their limits and brought to their knees by additional requirements for student models, curriculum sequences, and hypertext interfaces. Even though these computer simulations were forced to operate at the edge of their acceptability, an official Army evaluation verified the many benefits of simulation-based training systems (Farr & Psotka, 1992).

Depending on the level that a simulated device has been decomposed to, and the degree of learner response regarding manipulations and ensuing ramifications, feedback could attain various levels of detail (criticism 1, feedback specificity). Furthermore, as simulations become typically very reactive to learner actions, they can serve as a direct challenge to the second criticism (nonadaptability). Simulations, similar to discovery worlds, also leave quite a bit of freedom to explore and manipulate simulated objects and devices (countering criticism 4, restriction environments). However, the drawback of these systems is that a solid theoretical basis is lacking (criticism 3, atheoretical foundation). Simulation research in the 1980s spurred later work that attempted to incorporate pedagogical strategies into the simulation-based systems. Moreover, related developments continue to evolve in complexity with the addition of virtual-reality interfaces to three-dimensional models and simulations (Acchione-Noel & Psotka, 1993).

Two other areas of research and development gained prominence at this time: natural language processing (NLP) and authoring shells. While these research spheres were important in relation to ITS research, they could be applied within a variety of tutor types. For example, NLP could be used to communicate information to the learner (or accept input from the learner) in model-tracing tutors, discovery worlds, and so forth. And authoring shells could be built for the development of a range of tutoring systems. Because of this openness, the following two ITS-related issues won't be discussed in relation to our four criticisms, listed earlier.

19.4.2.7. Natural Language Processing (NLP). This technology was an important part of ITS right from the beginning. SOPHIE, in fact, was built on a powerful and original NLP technique developed by Richard Burton; it was called *Semantic Grammar.* Representing a powerful combination of carefully selected keywords with algorithms that searched the context for meaningful variables and objects, it worked surprisingly well, given its relative simplicity. Since communication is such an important element of ITS (see Wenger, 1987, for emphasis), it is not surprising that NLP technologies have been used in several ITS for discourse networks (Woolf, 1988) and especially for language instruction (Yazdani, 1990; Psotka, Holland & Kerst, 1992). The development of powerful, efficient Prolog compilers and languages on PCs has led to the implementation of some interesting instructional grammars that can handle discourse in English or other languages, and provide multimedia instruction in advanced language concepts and grammar, as well as simple vocabulary and verb declension. The potential addition of animations and immersion into virtual environments adds a bright new prospect to the old goal of *immersive* language learning.

19.4.2.8. Authoring Systems. The creation of computer-based environments to facilitate the design and development of ITS has been an important and continuing thread of research. The goal of authoring systems is to give relative computer novices a software toolkit to take advantage of the power of computers for designing instruction. An example of one powerful graphic authoring system developed over the last decade is that by Towne and Munro (1992).

Quite powerful CBT systems have been made available over the years. Research, beginning in the 1980s, attempted to adapt such systems as authoring shells for developing ITS. Miller and Lucado (1992) were among the first to integrate the power of CBT authoring environments with the technology of ITS. Their prototype system was the harbinger of many more powerful combinations of traditional CBT and next-generation ITS technologies. Most recently, DARPA has funded a unique consortium of Apple Computer, textbook publishers such as Houghton-Mifflin, and ITS experts Beverly Woolf and John Anderson to begin the development of next-generation authoring tools for instruction and training

The relative quiescence of the 80s transitioned into the current state of ITS affairs, marked by a perception of instability and controversy.

19.4.3 1990s: Great Debates

The four hot ITS topics right now may be broadly characterized as: (a) How much learner control should be allowed in systems? (b) Should learners interact with ITS individually or collaboratively? (c) Is learning situated, unique, and ongoing, or is it more symbolic, following from an information-processing model? (d) Does virtual reality (VR) uniquely contribute to learning beyond CAI, ITS, or even multimedia? There are, of course, proponents and opponents to each of these positions.

19.4.3.1. Degree of Learner Control. The debate over the amount of learner control that should be a part of the learning process has raged for many years (see 7.4.6, 12.2.3.5, 14.6.2, 22.5.5, 23.7.2, Chapter 33). On the one hand, some have argued that discovering information on one's own is the best way to learn (e.g., Bruner, 1961). On the other hand, structure and direction have been stressed as the important ingredients in the promotion of student learning (e.g., Ausubel, 1963). The same debate has appeared in the ITS arena. Two differing perspectives, representing the ends of this continuum, have arisen in response to the issue of the optimal ITS learning environment. One approach is to develop a computerized environment containing assorted tools, and allow learners freedom to explore and learn independently (e.g., Collins & Brown, 1988; Shute, Glaser & Raghavan, 1989; White & Horowitz, 1987). Advocates of the opposing position argue that it is more effective to develop straightforward learning environments with no digressions permitted (e.g., Anderson, Boyle & Reiser, 1985; Corbett & Anderson, 1989; Sleeman, Kelly, Martinak, Ward & Moore, 1989). This disparity between perspectives becomes more complicated because the issue is not just which is the better learning environment, but which is the better environment for whom, a classic aptitude-treatment interaction question (see 22.3.3; Cronbach & Snow, 1981). There are, undoubtedly, temporal aspects to this issue as well. For instance, it may be more efficient to learn a new cognitive skill initially by direct instruction, then later by greater exploration. In this way, learners can better control their own learning process.

Merrill, Reiser, Ranney, and Trafton (1992) investigated how human tutors dealt with the issue of learner control. They compared human- to computer-tutoring techniques, and found that, while expert human tutors did sometimes act like model tracers, they actually maintained a "delicate balance" between (a) allowing students freedom and control and (b) giving students sufficient guidance. In general, pedagogical research findings differ with regard to the amount of learner control to allow in automated systems (e.g., Fox, 1991; Lepper, Aspinwall, Mumme & Chabay, 1990; Merrill, Reiser & Landes, 1992). In addition to the temporal factor cited above, this issue of learner control is also greatly dependent on other variables, such as the subject matter being instructed, the desired knowledge or skill outcome, incoming aptitudes, and so on (see Kyllonen & Shute, 1989, for a complete discussion of these interacting variables). That is, if the desired learning outcome is a smoothly executed skill, it may be more efficient to instruct certain learning tasks with direct instruction and plenty of practice. But if the desired learning outcome is a functional mental model of relevant principles, an exploratory environment (complete with various components such as online circuits, ammeters, and resistors) perhaps is optimal to achieve that educational objective.

Most current computer-administered instructional systems do not foster self-reliance in students or encourage them to seek new information on their own. To rectify this deficit, Barnard, Erkens, and Sandberg (1990) propound the building of more flexible systems packaging communication expertise as a separate component. With less learner initiative, it's much easier to interpret input, but at what cost to learning outcome? In Japan, research is being conducted along these lines. The concept and development of ITS is becoming merged with interactive learning environments (ILE) to produce what is referred to as a "bi-modus learning environment" (BLE) (Otsuki, 1993). Whereas the main strength of ITS is its ability to derive a student model based on the identification of acquired rules, its main weakness is the inability to help learners acquire new knowledge by themselves. In contrast, students in an ILE can extract and comprehend rules induced from a complex domain, but the ILE cannot explicitly identify a student's misconceptions or tutor them in terms of their comprehension level. Thus the two (ITS and ILE) are complementary to one another, and BLE represents combining the strengths of each.

Another way to increase learner control has been suggested by Bull, Pain, and Brna (1993). Their intriguing alternative to traditional student modeling (that of replacing the burden of the ITS) is to produce accurate representations of the learner's knowledge state; the learner is empowered with greater control, e.g., to construct and repair the model. Bull and associates contend that their model will result in a more accurate representation of the learner's beliefs, and thus be more highly regarded by the student. The learner is expected to benefit through the reflection necessary to accomplish this modeling task. Unfortunately, no data have yet been collected about the efficacy of this novel approach.

"Coached practice environments" (i.e., Sherlock I and II) represent yet another way to provide control during learning by combining apprenticeship training with intelligent instructional systems (Lajoie & Lesgold, 1992; Lesgold, Eggan, Katz & Rao, 1992; see 7.4.5). These systems support greater learner initiative because the apprentice learns by doing (singularly or collaboratively); knowledge is anchored in experience; and the coach provides knowledge within an applicable context. Intelligent systems are developed with many of the characteristics of human apprenticeships, and performance can be easily assessed. Through replay and comparisons with the expert performance, this approach also supports trainee analysis of performance.

Salomon (1993) supports the trend of moving away from building traditional ITS and towards the design of systems as cognitive tools. He sees cognitive tools manipulated by students as instruments that promote constructive thinking, transcending cognitive limitations, and making it possible for students to engage in cognitive operations they wouldn't otherwise have been capable of. Some ITS programs make most diagnostic and tutorial decisions for the student; therefore they are not really cognitive tools because "they are not designed to upgrade students' intelligent engagements" (p. 180). Also in accordance with the notion of computers as learning tools, learners should have the option to alter the degree of control themselves, from none (e.g., didactic environment) to maximum (e.g., discovery environment), as necessary.

By shifting toward increased learner control, are individuals who are not very active or exploratory by nature being penalized or handicapped? Shute and Glaser (1990) investigated individual differences in learning from a discovery environment (Smithtown) and found that individuals who demonstrated systematic, exploratory behaviors (e.g., recording baseline data, limiting the number of changed variables) were significantly more successful in Smithtown compared to those who revealed less systematic behaviors. On the basis of that finding, they hypothesized in a different study (using an electricity tutor) that high-exploratory individuals would learn more from an inductive environment (than from a more directed, applied environment), and less-exploratory learners would benefit from a supportive, applied environment (compared to an inductive one). A person's exploratory level was quantified based on certain indices (e.g., number of tries and length of time spent changing a resistor value, using the on-line voltmeter or ammeter). Subjects were randomly assigned to one of two learning environments, and the data were analyzed, post hoc. The hypothesized learning style by aptitude treatment interaction (ATI) (see 22.3.3) was supported by the data (Shute, 1993b). So, discovery learning environments do not suit everyone equally well. For some, they provide a really bad fit. To determine whether this kind of learner style by treatment interaction is replicable, Shute (1994) conducted a *confirmatory* test of the same ATI, reported above. Subjects were placed a priori in one of two environments based on the decision rule obtained from the previous study. And, in fact, the ATI was confirmed (see 11.4.4, 22.3.6, and 33.6 for more on ATIs).

In conclusion, a midpoint between too much and too little learner control is probably the best bet as far as optimal ITS learning environment. Furthermore, this milestone should not be fixed, but should change in response to learners' evolving needs. Finally, learners should have some input into the design of the environment, as well.

Our next debate addresses the issue of whether learning alone is better or worse than learning in conjunction with others where "others" may mean other human beings, or with a computer acting as a "partner" in the learning process). As with everything relating to learning, there is probably no clear-cut answer to this question: Is there no "overall" superior way to learn? Rather, it is almost certain that interactions exist, where solo learning may be superior for certain topics (e.g., rote memorization of multiplication tables) or for particular learner types (e.g., highly motivated individuals). Collaborative learning may be more effective for other domains or persons. While we don't specifically address these interactions in the following discussions, they should be kept in mind (also refer to Chapter 6, 7.4.8, 23.4.4, Chapter 35).

19.4.3.2. Individual vs. Collaborative Learning. Traditionally, ITS have been designed as single-learner enterprises. Bloom (1984) and others have presented compelling evidence that individualized tutoring (using human tutors) engenders the most effective and efficient learning across an array of domains (see also Shute & Regian, 1990; Woolf, 1988). Furthermore, intelligent tutoring systems *epitomize* this principle of individualized instruction. In his often-cited 1984 paper, Bloom presented a challenge to instructional researchers that has been called the "two sigma problem." The goal is to achieve two standard-deviation improvements with tutoring over traditional instruction methods. So far, this goal has yet to be attained using individualized ITS.

An alternative approach to individualized instruction is collaborative learning, the notion that students, working together, can learn more than by themselves, especially when they bring complementary, rather than identical, contributions to the joint enterprise (Cummings & Self, 1989). Collaboration is defined as a process by which "individuals negotiate and share meanings relevant to the problem-solving task at hand" (Teasley & Roschelle, 1993, p. 229), and is distinct from cooperation, which relates to the division of labor required to achieve some task.

Two empirical questions relevant to this chapter include: (a) Are two heads better than one? (b) Can intelligent computer systems support collaborative learning endeavors? Recently, research is beginning to shed light on both of these questions. For example, many researchers have shown impressive student gains in knowledge and skill acquisition from collaborative learning environments (e.g., Brown & Palincsar, 1989; Lampert, 1986; Palincsar & Brown, 1984; Scardamalia, Bereiter, McLean, Swallow & Woodruff, 1989; Schoenfeld, 1985). Furthermore, the few studies of the effectiveness of collaborative learning in computer-based learning environments have also been positive (e.g., Justen, Waldrop & Adams, 1990; Katz & Lesgold, 1993; Papert, 1980).

There are basically two ways of implementing collaborative learning environments using computers: (a) A small group of learners interact with a single intelligent computer system, or (b) the computer system itself serves as the "partner" in the collaboration. The first way (i.e., a small group using one computer) represents an extension of the research on collaborative learning in classrooms. In this case, some of the issues that need to be addressed have been outlined by Teasley and Roschelle (1993). The system must be able to: (a) introduce and accept knowledge into a joint problem-solving space, (b) monitor ongoing activities for evidence of divergences in meaning, and (c) repair divergences that impede the progress of the collaboration. The difference between this list and general modeling issues in ITS is that it deals with a student model that's built on a joint, rather than single, problem-solving space. The second way of implementing collaboration (i.e., assigning the computer as the learner's partner) represents an intriguing twist on the notion of collaborative learning. To illustrate, Cummings and Self (1989) proposed a collaborative intelligent education system (IES) that engages the learner in a partnership. Here, the computer serves as a collaborator, not as an authoritarian instructor. In both cases, a student model still must be derived, either that of an individual or a group.

Additional research and controlled studies must be conducted in order to test the relative efficacy of collaborative versus individualized instruction. For a variety of reasons (e.g., greater range of shared knowledge, resource limitations, etc.), the notion of collaborative learning environments is appealing. There are a lot of unanswered research questions that need to be addressed, however. Some of these (listed in Katz & Lesgold, 1993) include: What parts of the curriculum should be learned collaboratively, and what parts learned individually? What teaching methods should be used to achieve the instructional goals, and how should they be sequenced to optimize learning? What should the computer tutor do while students work on problems? What additional roles could the computer coach perform? This area of research is also likely to shed light on the interactions mentioned earlier. We now present the third hot topic, namely, the nature of learning and its impact on ITS design.

19.4.3.3. Situated Learning Controversy. To supporters, this is not just a trend but a radically new perspective (or philosophy) that supports the integration of ". . . psychological theories of physical and cognitive skills, uniting emotions, reasoning, and development, in a neurobiologically grounded way" (Clancey, 1993, p. 98). It has also been referred to in the literature as "situated action" and "situated cognition." Recently, several prominent journals have devoted entire issues to the debate concerning the value of situated learning compared to the more standard paradigms (e.g., ACT*, SOAR): 1993 *Cognitive Science 17* (1), and 1993 *Journal of Artificial Intelligence and Education 4* (1) (see 12.3.1.2, 8.7, 8.9).

Obviously, one's belief in either situated cognition or the traditional information-processing model has implications for the design of ITS. To illustrate this distinction, first consider Greeno's summary of situated cognition's perspective on where knowledge resides: "Rather than thinking that knowledge is in the minds of individuals, we could alternatively think of knowledge as the potential for situated activity. On this view, knowledge would be understood as a relation between an individual and a social or physical situation, rather than as a property of an individual" (Greeno, 1989, p. 286). Next, consider the nature of knowledge from the information-processing perspective. Anderson's (1983) ACT* theory proposed two fundamental forms of knowledge: *procedural,* represented in the form of a production system, and *declarative,* represented in the form of a node-link network of propositions (see 5.4, 29.2). Both representations are believed to operate within long-term and short-term memory structures.

These two positions present quite different views on how learning, or knowledge acquisition, occurs. In the first case (situated cognition), learning is a process of creating representations, inventing languages, and formulating models *for the first time.* Learning is ongoing, occurring with every thought, perception, and action, and is situated in each unique circumstance. Situated cognition argues for an instructional system rich with explicit tools and varied exemplars that can support and extend learners' discovery processes. "Insight is more likely when the problematic situation is so arranged that all necessary aspects are open to observation" (Bower & Hilgard, 1981, p. 319).

The second position (information processing) sees learning as progressing from declarative knowledge, to procedural skills, to automatic skills, dependent on: enablers (i.e., what one already knows and can transfer to new situations) and mediators (i.e., cognitive processes determining what one can acquire, such as working-memory capacity and information-processing speed; e.g., Anderson, 1983, 1987; Kyllonen & Christal, 1990). Thus, learning refers to the addition and restructuring of information to a database, in accordance with specific learning mechanisms (e.g., knowledge compilation, transfer). To facilitate learning, one must build a system that can (a) analyze the initial state of knowledge and skill, (b) describe the desired or end state of knowledge and skill (learning outcome), and (c) present material and problems that will transition a learner from initial to desired state. This kind of tutoring system is based on a well-defined curriculum that's been so arranged to promote knowledge/skill acquisition (or facilitate transition from current to goal state).

It may be that these two positions are mutually exclusive. That is, knowledge either resides internally in one's head or externally in the environment. Alternatively, it may be that there is some overlap, whereby some forms of knowledge are stored and some are derivable from the current situation. In a preliminary attempt to bridge the gap between situated- and traditional-learning models, Shute and Gawlick-Grendell (1994) have recently developed a series of statistics modules, *Stat Lady.* Learning is situated within various gaming environments (e.g., "Stat Craps"). The theoretical postulates are that learning is a constructive process, enhanced by experiential involvement with the subject matter, that is, situated in real-world examples and problems. Furthermore, the system has a well-defined curriculum in accordance with popular learning theory.

According to constructivism, learners actively construct new knowledge and skills, either from what they already know (information-processing premise) or from what resides in the environment (situated cognition stance; see Chapter 7). Both positions would probably agree that learners do not come to a learning situation with a *tabula rasa,* but rather as active-pursuers (not passive-recipients) of new knowledge (e.g., Bartlett, 1932; Collins, Brown & Newman, 1989; Drescher, 1991; Edelman, 1987; Piaget, 1954). Both positions also support the premise that the construction process can be enhanced by environments supporting experiential learning. Research in this area has shown that knowledge derived experientially tends to be more memorable than passively received knowledge, because the experience ("doing" rather than "receiving") provides cognitive structure, and is intrinsically motivating and involving (e.g., Friedman & Yarbrough, 1985; Harel, 1991; Harel & Papert, 1991; Shute & Glaser, 1991; Spencer & Van Eynde, 1986). Finally, when instruction is situated (or anchored) in interesting and real-world problem-solving scenarios, that also is believed to enhance learning (Brooks, 1991; Brown, Collins & Duguid, 1989; Clancey, 1992; Collins, Brown & Newman, 1989; Lave & Wenger, 1991; Suchman, 1987; The Cognition & Technology Group at Vanderbilt, 1992).

The Cognition and Technology Group at Vanderbilt (1992) has also been working on developing a pedagogical approach to situated cognition. They define "anchored instruction" as an attempt to actively engage learners in the learning process by situating instruction in interesting and real-world problem-solving environments. Rather than teaching students how to solve particular problems, these systems teach generalizable skills, helpful across a variety of problem-solving situations. The major goal of this type of instruction is to create authentic-feeling environments in which one can explore and understand problems and opportunities experienced by experts in a domain, and learn about the tools these experts use. This group has developed a series of adventures for middle-school students focusing on math problem formulation and problem solving. These are the "Adventures of Jasper Woodbury" series (see 23.5.1.1). The goal of the project is to facilitate broad transfer to other domains, embodying several design principles: (1) video-based presentation, (2) narrative format, (3) generative learning, (4) embedded data design, (5) problem complexity, (6) pairs of related adventures, and (7) links across the curriculum.

One of the major problems with this whole debate over situated cognition versus traditional information-processing models is that the former position simply has not tested its underlying hypotheses at this time, while the latter has enjoyed decades of solid research. Vera and Simon (1993), rebutting Clancey's support paper(s) for situated learning, stated, "Clancey leaves us with philosophy (whether correct or not is another matter), but with precious little science" (p. 118). And that appears to be true. Because cognitive psychology is an empirical science, studies need to be conducted that examine claims made by any new position (see 42.5.2). For instance, supporters of our final "hot topic" of the 90s (virtual reality, or VR) claim that this new technology can improve learning by virtue of fully immersing the learner in the learning process (learning by saturation). But is there any veracity to this claim? It is certainly testable. The relationship between experience, learning, and pedagogy is a briar patch of thorny questions. Recent theoretical harangues on the nature of situated learning have laid a kind of groundwork for VR by arguing for an epistemology of learning based on experience.

19.4.3.4. Virtual Reality and Learning. A collection of technologies, known as virtual reality (VR), has recently been exciting the instructional-technology community. This new technology refers collectively to the hardware, software, and interface technologies available to the user interested in experiencing certain aspects of a simulated three-dimensional environment. The simulated aspects of the environment ("world") currently include a stereoscopic, low-to-medium fidelity visual representation displayed on a head-mounted display system. Using head-tracking technologies, one can update the display in accordance with head and body motions. This feature, along with the stereo disparity of the images on the two screens (one for each eye), supports the illusion of moving around in three-dimensional space (for more, see Chapter 15).

Unquestionably, VR changes the relationships between learning and experience, highlighting the role of perception (particularly visual) in learning. Experience is both social and perceptual, and VR epitomizes the notion of experiential learning. Many systems are now being developed that have demonstrated the success of the experiential approach. The current question is: Does VR represent the next logical, developmental step in the design of instructional systems? In other words, does the immersion experience (i.e., extra fidelity and related cost) significantly improve learning and performance beyond the more traditional pedagogical approaches?

Recently, there have been some empirical data collected on the relative success of VR in terms of instructional effectiveness as well as skill transfer to the real world. For instance, Regian, Shebilske, and Monk (1992) showed that people can, indeed, learn to perform certain tasks from virtual environments (e.g., console operations and large-scale spatial navigation). Next, knowledge and skill acquired in a VR have been shown to transfer to performance in the real world. Regian, Shebilske, and Monk (1993) found that: (a) VR console operations training can transfer/facilitate real-world console operations performance; and (b) VR spatial navigation training successfully transfers to real-world spatial navigation. In contrast to the Regian et al. (1993) findings, however, those reported by Kozak, Hancock, Arthur, and Chrysler (1993) showed *no* evidence for transfer of a "pick and place" task from VR to the real world. However, the criterion task used in that study was quite easy; thus, the conclusions may actually be inconclusive. So, even with the relatively poor fidelity and interface currently available in VR technology, there is some evidence for its efficacy and potential as a serious learning/training environment (see 15.8).

Another positive example of VR's potential for training was presented by Psotka (1993), who argued that VR creates one uniform point of view on any representation that overcomes the conflicts and cognitive load of maintaining two disparate points of view (Sweller, 1988). The reduced cognitive overhead resulting from the single "egocenter" in a VR should expedite information access and learning (see 15.6). Central to this perceptual experience of VR is the poorly understood phenomenon of immersion or presence. Preliminary insight based on the SIMNET experience (Psotka, 1993) provides not only personal testimonials to the motivating and stimulating effects of the social and vehicle-based immersion of synthetic environments but also preliminary effectiveness data on its potency for learning and training. That is, even though SIMNET provides an impoverished perceptual simulation of a tank in action, the cues from interactive communications among crew members, as well as the auditory and visual cues of the simulated sights, provide gut-wrenching and sweaty believability. What's more, the evidence clearly shows a level of training effectiveness (even without a curriculum) that is superior to many other classroom- and simulation-based efforts (Bessemer, 1991). Research is continuing on how to make this training more effective by including surrogate crew members and intelligent semiautomated forces in the environments. The need to involve dismounted infantry, not just tanks and vehicles, is creating a research base for better computational models of agents and coaches (Badler, Phillips & Webber, 1992).

Virtual reality shows promise in the construction of microworlds for physics and other science instruction. For instance, Loftin and Dede (1993) are creating a Virtual Physics Laboratory from the base facilities of a VR world created for NASA astronaut training. In their virtual laboratory, students can conduct experiments in a virtual world where everyday accidents, structural imperfections, and extrinsic forces, such as friction, can be completely controlled or eliminated. Balls that bounce with complete determinism can be measured accurately at all times and places, and can even leave visible trails of their paths. The effects of gravity can be controlled, and variations of gravity can be experienced visually, and, perhaps, even kinesthetically.

Although the perceptual aspects of experience are clearly important, it is easy to assume that there are no difficulties to learning from existing visual representations and simulations, like photographs, graphs, and static drawings. It is easy to downplay and overlook difficulties in modern learning environments. Most of us are experts at interpreting visual representations on printed pages (figures, graphs, photographs, icons, drawings, and prints), but it's easy to forget the difficulty we once experienced as we tried to interpret scatterplots and line graphs. We know from many studies that those difficulties never completely go away. For younger learners, they may be even more pronounced. VR can remove these difficulties to a degree and make information more accessible through the evolutionarily prepared channels of visual and perceptual experience. As to the question of whether the delivered "bang" is worth the bucks, the jury is still out.

We now turn our attention away from these controversies, and toward the analysis of a collection of ITS that have been systematically evaluated and reported in the literature. The purpose of this section is to provide a flavor for evaluations that have been conducted, rather than to review all possible evaluations.

19.5 ITS EVALUATIONS

> Building a tutor and not evaluating it is like building a boat and not taking it in the water. We find the evaluation as exciting as the process of developing the ITS. Often, the results are surprising, and sometimes they are humbling. With careful experimental design, they will always be informative (Shute & Regian, 1993, p. 268).

Which systems instruct effectively? What makes them effective? One might think that increasing the personalization of instruction (e.g., model tracing) would enhance learning efficiency, and in the process improve both the rate and quality of knowledge and skill acquisition. But results cited in the literature on learning, in relation to increased computer adaptivity, are equivocal. In some cases, researchers have reported no advantage of error remediation in relation to learning outcome (e.g., Bunderson & Olsen, 1983; Sleeman, Kelly, Martinak, Ward & Moore, 1989). In others, some advantage has been reported for more personalized remediation (e.g., Anderson, Conrad & Corbett, 1989; Shute, 1993a; Swan, 1983).

If, however, more researchers conducted controlled ITS evaluations, this issue would be easier to resolve. But, in addition to the availability of relatively few reported evaluations of ITS, there has been little agreement on a standard approach for designing and assessing these systems (see 39.5). Results from six ITS evaluations will now be presented.

19.5.1 Six ITS Evaluations

A few examples of systematic, controlled evaluations of ITS reported in the literature include: the LISP tutor (e.g., Anderson, Farrell & Sauers, 1984), instructing LISP programming skills; Smithtown (Shute & Glaser, 1990, 1991), a discovery world that teaches scientific inquiry skills in the context of microeconomics; Sherlock (Nichols, Pokorny, Jones, Gott & Alley, 1995; Lesgold, Lajoie, Bunzo & Eggan, 1992), a tutor for avionics troubleshooting; Bridge (Bonar, Cunningham, Beatty & Weil, 1988; Shute, 1991), teaching Pascal programming skills; *Stat Lady*, instructing statistical procedures (Shute & Gawlick-Grendell, 1993), and the Geometry tutor (Anderson, Boyle & Yost, 1985), providing an environment in which students can prove geometry theorems. Results from these evaluations show that these tutors *do* accelerate learning with, at the very least, no degradation in outcome performance compared to appropriate control groups.

19.5.1.1. The LISP Tutor. Anderson and his colleagues at Carnegie-Mellon University (Anderson, Farrell & Sauers, 1984) developed a LISP tutor that provides students with a series of LISP programming exercises and tutorial assistance as needed during the solution process. In one evaluation study, Anderson, Boyle, and Reiser (1985) reported data from three groups of subjects: human-tutored, computer-tutored (LISP tutor), and traditional instruction (subjects solving problems on their own). The time to complete identical exercises were: 11.4, 15.0, and 26.5 hours, respectively. Furthermore, all groups performed equally well on the outcome tests of LISP knowledge. A second evaluation study (Anderson, Boyle & Reiser, 1985) compared two groups of subjects: students using the LISP tutor and students completing the exercises on their own. Both received the same lectures and reading materials. Findings showed that it took the group in the traditional instruction condition 30% *longer* to finish the exercises than the computer-tutored group. Moreover, the computer-tutored group scored 43% *higher* on the final exam than the control group. So, in two different studies, compared to traditional instruction, the LISP tutor was apparently successful in promoting faster learning with no degradation in outcome performance.

In a third study using the LISP tutor to investigate individual differences in learning, Anderson (1990) found that when prior, related experience was held constant, two "metafactors" emerged. These two metafactors, or basic learning abilities, included an *acquisition* factor and a *retention* factor. Not only did these two factors explain variance underlying tutor performance, they also significantly predicted performance on a paper-and-pencil midterm and final examination.

A fourth study with the LISP tutor concerns the usefulness of productions for analyzing learning. In analyzing student performance on the first six problems in Chapter 3 of the LISP tutor, Anderson (1993, p. 32) discovered uneven, unsystematic trends in learning. One problem was relatively easy, and the next might be relatively more difficult. However, by decomposing the problems into their constituent production rules, Anderson was able to convert the chaos of these results into very systematic program solution learning curves, for both time and accuracy. He analyzed performance on individual production rules across problems. Because productions were reused, and others newly introduced in each problem, he could plot performance in terms of the number of opportunities each production rule had for contributing to an additional unit of LISP code. This simplifying transformation demonstrates that knowledge is acquired in terms of production rules, and that if we are to understand how learning cognitive skills is to be explained, our analysis of the task and data ought to be conducted in terms of production rules.

19.5.1.2. Smithtown. Shute and Glaser (1991) developed an ITS designed to improve an individual's scientific inquiry skills within microworld environment for learning principles of basic microeconomics. In one study (Shute, Glaser & Raghavan, 1989), three groups of subjects were compared: a group interacting with Smithtown, an introductory economics classroom, and a control group. The curriculum was identical in both treatment groups (i.e., laws of supply and demand). Results showed that while all three groups performed equivalently on the pretest battery (around 50% correct), the classroom and the Smithtown groups showed the same gains from pretest to posttest (26.4% and 25.2%, respectively); they significantly outperformed the control group. Although the classroom group received more than twice as much exposure to the subject matter as did the Smithtown group (11 vs. 5 hours, respectively), the groups did not differ on their posttest scores. These findings are particularly interesting because the instructional focus of Smithtown was not on economic knowledge, per se, but rather on general scientific inquiry skills, such as hypothesis testing.

19.5.1.3. Sherlock. "Sherlock" is the name given to a tutor that provides a coached practice environment for an electronics troubleshooting task (Lesgold, Lajoie, Bunzo & Eggan, 1990). The tutor teaches troubleshooting procedures for problems associated with an F-15 manual avionics test station. The curriculum consists of 34 troubleshooting scenarios with associated hints. A study was conducted evaluating Sherlock's effectiveness using 32 trainees from two separate Air Force bases (Nichols, Pokorny, Jones, Gott & Alley, 1995). Pre- and posttutor assessment used verbal troubleshooting techniques as well as a paper-and-pencil test. Two groups of subjects per Air Force base were tested: (1) subjects receiving 20 hours of instruction on Sherlock, and (2) a control group receiving on-the-job training over the same period of time. Statistical analyses indicated that there were no differences between the treatment and the control groups on the pretest (means = 56.9 and 53.4, respectively). However, on the verbal posttest as well as the paper-and-pencil test, the treatment group (mean = 79.0) performed significantly better than the control group (mean = 58.9) and equivalent to experienced technicians with several *years* of on-the-job experience (mean = 82.2). The average gain score for the group using Sherlock was equivalent to almost 4 years of experience.

19.5.1.4. Pascal ITS ("Bridge"). An intelligent programming tutor was developed to assist novice programmers in their designing, testing, and implementing Pascal code (Bonar, Cunningham, Beatty & Weil, 1988). The goal of this tutor is to promote conceptualization of programming constructs or "plans" using intermediate solutions. A study was conducted with 260 subjects who spent approximately 12 hours learning from the Pascal ITS (see Shute, 1991). Learning efficiency rates were estimated from the time it took subjects to complete the curriculum. This measure involved both speed and accuracy, since subjects could not proceed to a subsequent problem until they were completely successful in the current one. To estimate learning outcome (i.e., the breadth and depth of knowledge and skills acquired), three criterion posttests were administered measuring retention, application, and generalization of programming skills.

The Pascal curriculum embodied by the tutor was equivalent to about half a semester of introductory Pascal. That is, the curriculum equaled about 7 weeks or 21 hours of instruction time. Adding 2 hours per week for computer laboratory time (conservative estimate), the total time spent learning a half-semester of Pascal the traditional way would be at least 35 hours. In the study discussed above, subjects completed the tutor in considerably less time (i.e., mean = 12 hours, SD = 5 hours, normal distribution). So, on average, it would take about 3 times as long to learn the same Pascal material in a traditional classroom and laboratory environment as with this tutor (i.e., 35 vs. 12 hours).

While all subjects finished the Pascal ITS curriculum in less time compared to time needed to complete the curriculum under traditional instructional methods, there were large differences in learning rates found at the end of the tutor. For these subjects (having no prior Pascal experience), the maximum and minimum completion times were 29.2 and 2.8 hours, a range of more than 10:1. In addition, while all 260 subjects successfully solved the various programming problems in the tutor's curriculum, their learning outcome scores reflected differing degrees of achievement. The mean of the three criterion scores was 55.8% (SD = 19, normal distribution). The range from the highest to the lowest score, 96.7% to 17.3%, represented large between-subject variation at the conclusion of the tutor. To account for these individual differences in outcome performance, Shute (1991) found that a measure of working memory capacity, specific problem-solving abilities (i.e., problem identification and sequencing of elements), and some learning style measures (i.e., asking for hints and running programs) accounted for 68% of the outcome variance.

19.5.1.5. *Stat Lady*. Two studies have been conducted to date with *Stat Lady*. One study (Shute, Gawlick-Grendell & Young, 1993) tested the efficacy of learning probability from *Stat Lady* in relation to a traditional lecture and a no-treatment control group. Results showed that both treatment groups learned significantly more than the control group, yet there was no difference between the two treatment groups in terms of pretest to posttest improvements after 3 hours of instruction. The results were viewed as very encouraging because not only was the lecture a more familiar learning environment for these subjects but also the professor administering the lecture had more than 20 years experience teaching this subject matter, while this was *Stat Lady's* first teaching assignment. When test items were separated into declarative and procedural categories, they found that: (a) Students using *Stat Lady* acquired significantly more declarative knowledge than the other groups, but (b) when procedural skill acquisition was assessed, the lecture group prevailed. Finally, a significant aptitude-treatment interaction was obtained where high-aptitude subjects learned significantly more from *Stat Lady* than from the lecture environment, but for low-aptitude subjects, there was no difference in learning outcome by condition. Together, these results suggest that a teacher-computer combination maximizes learning.

The second study (Shute & Gawlick-Grendell, 1994) compared learning from *Stat Lady* vs. learning from a paper-and-pencil Workbook version of the identical curriculum, and addressed the question: What does the computer contribute to learning? Findings showed that *Stat Lady* learners performed at least as well (and in some cases, much better) on the outcome tests compared to the Workbook group, again despite the presence of factors strongly favoring the traditional condition. Specifically, they found that (a) *Stat Lady* was clearly the superior environment for high-aptitude subjects; (b) *Stat Lady* subjects acquired significantly more declarative knowledge than the Workbook subjects; and (c) regardless of aptitude, the majority of learners found the *Stat Lady* condition to be significantly more enjoyable and helpful than the Workbook condition.

19.5.1.6. Anderson's Geometry Tutor. The geometry tutor (Anderson, Boyle & Yost, 1985) provides an environment for students to prove geometry theorems. The system monitors student performance and jumps in as soon as a mistake is made. The skill this system imparts is how to prove geometry theorems that someone else has provided. Schofield and Evans-Rhodes (1989) conducted a large-scale evaluation of the tutor in place within an urban high school. Six geometry classes were instructed by the tutor (in conjunction with trained teachers), and three control geometry classes taught geometry in the traditional manner. The researchers closely observed the classes using the geometry tutor and traditional instruction for more than 100 hours. One of the really nice and intriguing results of Schofield and Evans-Rhodes's (1989) evaluation of this tutor was the counter-intuitive reversal of its effects. Although the geometry tutor was designed to individualize instruction, one of its pragmatic and unintended side effects was to encourage students to share their experiences and cooperatively solve problems. Since their experiences with the geometry tutor was so carefully controlled by the immediate feedback principles of its operations, the tutor guaranteed that students' experiences were much more uniform and similar than was the case for normal classrooms. As a result, students could more easily share experiences and make use of one another's experiences and problem-solving strategies. The practical result was a great deal of cooperative problem solving.

19.5.2 Conclusions from the Six Evaluation Studies

These evaluation results all appear very positive regarding the efficacy of ITS; however, there is always a selection bias involved with the publication of unambiguous evidence of successful instructional interventions. We are familiar with other (unpublished) tutor-evaluation studies that were conducted but were "failures." However, the general positive trend is viewed as encouraging, especially given the enormous differences among the six tutors in design structure as well as evaluation methods. The findings indicate that these systems do accelerate learning with no degradation in final outcome.

Obviously, principled approaches to both the design and evaluation of ITS are badly needed before we can definitively judge the merits of these systems. Some principled approaches are beginning to emerge. For example, Kyllonen and Shute (1989) outlined a taxonomy of learning skills that has implications for the systematic *design* of ITS. They hypothesized a multidimensional interaction predicting learning outcome as a function of type of learning/instructional environment, type of knowledge/skill being instructed, subject matter, and characteristics of the learner (e.g., aptitude, learning style). With a few modifications to this taxonomy, Regian and colleagues at the Armstrong Laboratory are currently trying to fill in the cells in the matrix through systematic, empirical studies designed to assess performance across a range of these aforementioned dimensions. Their goal is to map instructional and knowledge-type variables to learning.

In terms of systematic approaches to *evaluating* ITS, Shute and Regian (1993) suggested seven steps for ITS evaluation: (1) Delineate goals of the tutor. (2) Define goals of the evaluation study. (3) Select the appropriate design to meet defined goals. (4) Instantiate the design with appropriate measures, number, and type of subjects and control groups. (5) Make careful logistical preparations for conducting the study. (6) Pilot test tutor and other aspects of the study. (7) Plan primary data analysis concurrent with planning the study. These principles may also be employed as a framework for organizing, discussing, and comparing ITS evaluation studies.

19.6 FUTURE ITS RESEARCH AND DEVELOPMENT

> What is possible for the future includes ample computing resources for every student . . . tapping electronically many resources outside the classroom. It includes the idea of a personal factotum that could serve as a knowledgeable intermediary . . . to bridge the gap between the classroom and the external world. . . . Virtual field trips linking libraries and museums will have their holdings available in electronic (or photonic) form . . ."(Nickerson, 1988, p. 312).

We've seen where ITS research and development has been, and we've discussed a few of the systems that have been evaluated in controlled studies. We'll now examine some of the conceivable futures for these systems. Given the diversity of researchers in the area, and the great differences among learners, there will be, in reality, many different streams of research co-occurring, and the most likely future is probably a composite of them all.

19.6.1 Future 1: Immersive Learning Environments Evolve from ITS

> Alden (age 11) walks into his cubicle at school and excitedly puts on his VR bodysuit. Today's itinerary (jointly produced by Alden and his main teacher) is teeming with new learning adventures. After taking a Dramamine, he boards a boat heading up the Nile. This trip (and his on-line tour guide) will help him learn about East Africa's geography, flora, and fauna as he cruises, observes, hears, and smells things along the world's longest river. When the trip concludes, he plans on visiting Olduvai Gorge for some archeological excavations (after all, he's already in Africa). Specifically, Alden will get a chance to help dig out some early human remains. Then, for a change of pace, Alden and his VR pal Rafael, who lives in Mexico City, will meet in a happenin' space station they programmed together. They are learning each others' language and culture: Rafael speaks English and helps Alden learn Spanish, while Alden speaks Spanish and assists Rafael with his English. Following a real lunch (not a virtual one, as all this learning makes one hungry), Alden concludes his day on an artistic note. He's creating a VR masterpiece representing his interpretation of the classical score "The Wall," by the noted composer Roger Waters, designing virtual sculptures, their choreography, and musical arrangement.

This imagined future using "immersive learning environments" can attain its instructional goals as follows. As a ghost presence, the tutor in these new systems can interact with a student through digital speech, through text that floats in the air, or through replays. As an embodied presence, the tutor can vary in reality from a stick figure to a realistic mannequin, with facial expressions and voice. The possibilities for realistic guidance that is as believable and as forceful as a real tutor may be quite difficult to achieve, but it can be dramatic in implications. The believability of these new systems hinges on the quality of the immersive experience they provide. The differences between an immersive learning environment and its 2-D simulation counterpart depends on the results of immersion and in the different ways that students can interact with the world. Instead of moving a mouse or a joystick, learners can move their own hands to pick something up (see 8.5.2). Although they might not feel the object accurately, there are enough cues to provide the *sensation* of picking things up. First, they see it happening, and vision clearly dominates other senses to provide a compelling illusion. Contact and force can be provided realistically with expensive force-feedback devices, or suggestively with sounds, such as a ping that denotes collision or touching (see 13.5.4, Chapter 15, 29.5).

VR also opens the opportunity for providing handicapped or disabled people an experience of unfettered motion, or new interfaces to control the world with minimal movements. It can make invisible forces like gravity and air pressure visible and, hence, more comprehensible to students. For instance, Minstrell (1988) pointed out that high school students go through a period of misconceptions during which they confuse gravity and air pressure; so that when air is pumped out of a bell jar, objects inside it are expected to become lighter or even float. VR offers an opportunity for doing a set of experiments in which the forces of gravity and air pressure could be made visible through graphic icons, such as colored arrows or textures. As the gas is removed from a bell jar, it could be visible as a colored gas flowing out. Students could actually reach into the bell jar and manipulate the objects as the gas is removed. They could even adopt the point of view, or frame of reference, of an object *inside* the bell jar and experience the change in forces directly. Making these forces visible in a multitude of lifelike and believable environments may have profound effects on children's understanding of science.

It should be noted that the same problems that plague ITS are relevant to VR. That is, the emphasis needs to perhaps shift away from omnipotent VR systems, toward a collection of specific minisystems and goals (e.g. teach the knowledge of *X*, the skill of *Y*, and provide the kinesthetic feedback for *Z*).

19.6.2 Future 2: Traditional ITS Disappear; Specific Cognitive Tools Dominate

> Whitney (age 14) arrives in her classroom and takes a seat at her learning station, a large comfortable desk with an embedded computer. The touch screen is divided into many different areas that have distinct functions (e.g., graphics, spreadsheet, sound analyzer, dozens of databases). From the front of the class, a visiting detective (serving as the day's teacher) accesses the international police database (IPD) and obtains details surrounding a grisly murder that happened the previous month in a small Italian city. She electronically transmits all of the information to the students, which includes electronic photographs of the physical evidence (e.g., the body and the weapon), psychological profiles of the victim and 11 suspects, recorded interviews, alibis and motives, phone logs, and so on. The students have to engage in a variety of coordinated cognitive activities to solve the murder mystery. Whitney first brings up the psychological profile of the dead man. After reading the file, she notes in her electronic scratch pad that the victim had a history of drug abuse and depression. On another part of her 25-inch screen, she accesses a 3-D photo of the victim, zooms in on his arms, and sees evidence of two recent intravenous injections. The pathology report from the coroner's office concluded that the victim died from a gunshot wound to his heart, but traces of a narcotic substance were also found in his body. Playing the interview tapes on her "stress analyzer," Whitney discovers that two of the suspects are clearly lying. Throughout the day, puzzle pieces slowly come together, the detective-teacher offers a few suggestions, and, finally, Whitney figures out whodunit (with .93 probability of accuracy).

In this vision of the future, "omnipotent" intelligent tutoring systems have been replaced by collections of specialized educational or cognitive tools: technological devices that help people to perform cognitive tasks (i.e., help them know, think, or learn). For example, simulators, smart spreadsheets, and extensive databases are cognitive tools available within classrooms. Apprenticeship training is envisioned as the main source of imparting skill, in conjunction with the supplemental simulator and associated tools for the apprentice to employ during learning. The training situations relate to real-world events, thus placing learning within a meaningful context.

One reason that ITS may disappear in the future is that, while many researchers agree that intelligence in an ITS is directly a function of the presence of a student model, the student model may, in fact, be the wrong framework around which to build good learning machines. Derry and Lajoie (1993) presented six reasons why the student modeling paradigm is problematic: (1) In complex domains, the student model cannot specify all possible solution paths. (2) One cannot determine or induce all possible "buggy" behaviors. (3) "Canned" text is antithetical to principles of tutorial dialog. (4) Reflection and diagnosis should be performed by the *student*, not the tutor. (5) Implementing the student modeling approach is very difficult, technically. (6) Model tracing is only applicable to procedural learning, but the focus should be on critical thinking and problem solving.

A second factor that could contribute to the decline of ITS is that the term *intelligent tutoring system* is associated with philosophical issues relating to the nature of intelligence. Many people associate intelligence with awareness and, since no AI system could be said to have achieved awareness, these people would not grant that any ITS had ever been developed. Nevertheless, dozens of "intelligent" tutoring systems have been routinely reported in the literature, and even more discussed at conferences. So, the name (and hence, the whole enterprise) may be inappropriate or misleading. Simply put, ITS may promise too much, deliver too little, and constitute too restrictive a construct. Gugerty (1993) summed it up best:

> There is a sense in which the goals of traditional intelligent tutoring systems are both too ambitious and too narrow. Most traditional ITS . . . are designed to provide tutoring in a stand-alone setting. . . . This ambitious goal requires that the ITS handle all aspects of the very difficult task of tutoring, including expert problem solving, student diagnosis, tailoring instruction to changing student needs, and providing an instructional environment. . . . On the other hand, the goal of developing very intelligent stand-alone ITS is narrow in the sense that it limits our conception of how intelligence can be incorporated into computer-based training and education (p. 3).

As a parallel, consider what happened in the field of robotics. First-generation robots were constructed out of pure research curiosity. Then, after the initial flurry of excitement in the 1960s and early 1970s died down, emphasis shifted from building single-system robots to more emphasis on building component parts. This trade-off was due to the problems associated with designing a system that has general-purpose problem-solving skills versus one with more focused expertise. The next generation of robots, arising from the work being done on the individual parts, may resolve this conflict by becoming an expert in a given domain, but also possessing a wide repertoire of general problem-solving skills. The same applies for ITS. Rather than attempting to build an omnipotent tutor, a more fruitful approach might be to create a coherent collection of computerized tools (i.e., a divide-and-conquer strategy) (see 12.4, Chapter 24).

19.6.3 Future 3: Distance Learning

Curtis (age 9) rolls out of bed, greets his parents (already at work in their cubicles), eats breakfast, glances at the sleet falling outside, then ambles over to his computer for his morning curriculum. Curtis "goes to school" in his home. When he logs onto the Public School System, he first checks his mail, then receives a menu of options for the morning's learning project: Would he like to learn about *Tyrannosaurus Rex,* the politics leading up to World War II, or what caused the California earthquake of 1994? All he has to do is tap into the appropriate database, travel to the correct geographical region and time period, and interact with these respective environments through the multimedia systems. The respective databases all include on-line hosts to narrate events and answer questions, movies to depict a range of relevant topics (from mundane to crucial), and simulators to allow Curtis to experiment within the different worlds. After choosing *T. Rex* as his learning project, the host narrates some basic declarative information (e.g., when they existed and for how long, size of the dinosaur, diet, mating habits, other coexisting plants and animals), then Curtis uses the simulator to manipulate geological events to see their ramifications on the dinosaur. The first thing he does is to reverse the advancing Ice Age (introducing a global warming trend in its place), and then sees its implications not only on the survival of the lizard king but also on the evolution of other plants and animals on the planet. Periodically, the host asks for some predictions, Curtis responds, and receives feedback from the host. On occasion, other students in the same module communicate their findings and questions to him over the network lines.

As can be seen, this future is attractive for a lot of reasons. With distance learning, one can allow learners to stay at home or at some other convenient learning location (saving time and transportation costs) and connect to a rich network of information and training software, available across an information superhighway (see Chapter 13). To achieve this future, expert systems—spanning a huge array of possible domains—are needed that present comprehensive information, as well as provide thought-provoking questions, and respond to student-directed queries. The network should also allocate nodes to which one's peers can be connected, thus providing for collaborative learning opportunities. Notice that this distance-learning future is not limited to accessing declarative knowledge from databases. Rather, software (e.g., simulators) should also be accessible to practice skill in any specific domain.

In this future, it is possible to access quickly on-line, digital-rich *libraries* with virtually limitless realms/databases for our personal learning pleasure. And while the educational horizon will invariably include VR technology as an important instructional medium (see Future 1), it will be just one of many media.

Finally, to attain this future and the metaphor and promise of the library as a knowledge space (i.e., the epitome of Carbonell's dream and the hypertext vision), we must first make a fundamental change on how we think about education. Our narrow conception of education (e.g., "school"), only relevant for those between ages 5 and 18, is no longer appropriate. Education should be for everyone, all ages, and available in all places.

19.6.4 Future 4: Individualized Learning Is Out; Collaborative Learning Is In

Sierra, Nicole, Fernando, Sasha, Kevin, and Uri comprise "team 3." They are between the ages of 18 to 22 (college sophomores). In their sociology class, there are two professors and five teams, each team reflecting an optimal mixture of aptitude, gender, learning styles, personality types, and ethnic backgrounds. They are all geared up for their on-line VR lesson on "racial prejudice." The six students are transported to Birmingham, Alabama, on a hot August day in 1951. In reality, only Sasha and Kevin are African-American, but in this lesson, all six kids are transformed into "Negroes" (as they're called in 1951). The lesson requires them to take a city bus to a "Whites Only" park that has a nice public swimming pool, try to swim in the pool, then go home to their impoverished residences on the outskirts of town. Problems arise immediately in this compelling simulation when they board the bus. Automatically, they all sit down in the front seats; after all, there are only four other riders on the bus, sitting in the middle section. The white bus driver rudely informs them to "move to the back" whereupon Sierra (team 3's outspoken leader) politely asks "Why?" When she gets slapped for her impudence, Nicole starts to cry. But Sierra persists. Then the bus driver utters some very ugly sentiments about them all, based solely on their skin color. They see by his reddening face and posture that he's about to strike out again, so they collectively decide to move quickly to the back of the bus. During the ride to the park, they discuss their experiences (what they feel, what they could have done differently, what caused this state of affairs, etc.). Sasha and Kevin contribute valuable information to the discussion from personal tales related to them by their grandparents and great-grandparents. Finally they arrive at the park, and things really go downhill from there. They're not allowed to enter the park or swim in the pool, they're called "dirty" and worse, and the simulation makes them all painfully aware of racial prejudice. Afterward, team 3 reviews and discusses all of the events, and their professors provide information, as needed, about the historical roots of racial prejudice leading up to the situation they encountered in their lesson.

The motivating force driving this future is the belief that collaborative learning is superior to individualized learning (se also Chapter 35). That is, learning may be invaluably enhanced from conversations with those who have differing opinions, backgrounds, or skills; know more about some topic; or who can ask perceptive, thought-provoking questions. Basic research is being conducted in cognitive and social psychology that seeks answers to questions pertaining to the optimal compositions of learner groups. Some of these research questions include: Is it better to mix genders, or have more homogeneous groupings? When establishing groups based on aptitude levels, is it better to match highs with highs, or a high with a low? What are the optimal coordinations of affective characteristics (e.g., passive with gregarious)? And what other cognitive/social considerations should be made (e.g., letting individuals self-select their group vs. being assigned)? According to Resnick and Johnson (1988), sociological studies show that most people prefer personal sources of information, and computers can enhance such communications. For more information on this topic, see Shute, Lajoie, and Gluck (in press).

Technology is evolving to the point where computer systems can routinely contain learning environments that support a high level of social interaction. This important technology facilitates effective learning, especially within the classroom. The atmospheres in the classrooms containing the connected computerized environment are boisterously controlled, similar to what Feurzeig (1988) found in a collaborative mathematics course that was "... more like a beehive than a math class" (p. 117). These collaborative classrooms can even support networked VR, which means that students, trainees, and experts can interact between schools and remote sites, and that trainees and instructors can share the same experience. Learners can work collaboratively on the same project. On the other hand, different students can work on the same project at the same time, without awareness of each other's presence, but with some invisible instructor lurking over their shoulders. The number of combinations are staggering, and their learning/training potential is unknown.

The other person in the networked world could also be an autonomous agent, or cyborg, part real and part synthetic. This idea raises a whole new set of possibilities for a computer coach, explanations, and guidance. "Social interface agents" (Thorisson, 1993) have progressed steadily as information about how to direct gaze, when to use paraverbals (hmmm, uh . . .) and when to take turns in a dialogue, all become better understood. Improvements in modeling human actions and planning (e.g., Badler, Phillips & Webber, 1992), including natural language interaction, will soon lead to the development of virtual agents that can coach and guide learners' actions within carefully planned learning activities. Some of these interactions are already available in a text form (Curtis & Nichols, 1993). These virtual agents focus on students' errors by offering experts' stories (Kedar, Baudin, Birnbaum, Osgood & Bareiss, 1993). Networked digital spaces, such as digital libraries, demand new techniques for navigating through these complex spaces without getting lost. Issues of how to maintain a sense of location (Benedikt, 1991) and how to best use these environments to support memory with the method of loci (Neisser, 1987) need more research.

As shown in the above illustration, VR provides a new saliency on the notion that some things (such as race and gender) are constructed, and that we can become what we play, argue about, and build. For instance, text-based VR already invites the participation of women and girls in social interactions in ways that adventure games like dungeons and dragons did not (Turkle, 1993). Turkle points out that MUDs (i.e., multiuser dungeons) are easily used for gender swapping. When gender roles are switched, sexist expectations and overt demands that might be ignored in daily life become highly visible and reactive, and they are openly discussed. The MUD then becomes an evocative object for a richer understanding, not only of sexual harassment but also of the social construction of gender.

19.6.5 Future 5: The ITS Approach Continues; Becoming Truly Intelligent

> Ken (age 10) arrives at the math lab where he sits in front of a computer that is going to help him learn to solve algebra word problems better. Today's focus is on those troublesome distance-rate-time problems. After stating his name, the computer accesses Ken's records, flagging his salient strengths and weaknesses (i.e., not only his higher-level aptitudes but also the low-level productions that he's acquired and not yet acquired). Beginning with a review of concepts and skills that he learned the day before, the ITS generates a problem that is just a little bit out of his grasp. The ITS then works out the correct solution to the problem, along with an alternative solution that Ken is very likely to come up with based on its student model of him. In fact, he solves the problem exactly as the tutor predicted. As part of its student model of him, the ITS "knows" to instruct Ken with an emphasis on a graphical representation of the problem to clarify the discrepancy between the correct and incorrect solutions and facilitate the formation of a functional mental model. Thus, the tutor presents two animated trains appearing on opposite sides of the screen that converge at a point almost in the middle of the screen. They travel at different rates of speed. The problem statement stays up at the top of the screen, and the tutor points out, as it periodically pauses the simulation, what elements should be attended to and when. Ken states that he understands the mapping between the explicated mental model, the appropriate equation, and the relevant parts of the word problem. So the ITS presents an isomorphic word problem. This time he solves it correctly, without any supplemental graphics. Ken exercises an option to play around with some trains, missiles, and boats on his own for a while to test his emerging understanding. He views his "score" of curricular elements acquired, and seems a little frustrated about his progress, but the ITS reassures him that he is proceeding at a reasonable rate. Instruction and learning continue.

For ITS to evolve to the point seen in the above scenario, more controlled research must be conducted in three areas of intelligence: the domain expert, the student model, and the tutor. First, the subject matter must be understood by the computer well enough for the embedded expert to draw inferences or solve problems in the domain. Next, the system must be able to deduce a learner's approximation of that knowledge. Finally, the tutorial strategy must be intelligent to the point where an on-line tutor can implement strategies to reduce the differences between the expert and student performance (Burns & Capps, 1988).

Solutions to problems involving difficult AI, psychology, and pedagogy will emerge from research endeavors that yield information about effective and efficient ways to (a) represent, utilize, and communicate domain knowledge; (b) represent an individual's evolving knowledge state (for both declarative knowledge and procedural skill); and (c) instruct the material most effectively for a particular learner. Some specific research questions include: How can computers better understand natural language (input as well as output)? What kinds of inference mechanisms can optimally model students' knowledge status? How can computers be programmed to understand "semilogical" reasoning (including intuitions, pet theories, prior experiences)? What are the specific characteristics of learners who perform better in certain types of learning environments and not in others? Are certain domains better suited for specific instructional methods? When should feedback be provided, what should it say, and how best should it be presented? How much learner control should be allowed?

Some additional limitations of current ITS have already been mentioned (e.g., student models cannot specify all possible solution paths in complex domains; model tracing is only suitable for procedural learning). One possible solution would be to use a kind of model-tracing approach for instructing well-defined procedural skills, using an underlying expert and student model that are primarily rule based. And for instructing declarative information or complex, ill-structured domains, the ITS may include a knowledge base that is a semantic net with extensive indexing (like CBR).

Whatever future ultimately evolves from ITS, the fields of AI, education, and psychology have profited enormously from the contributions made in the ITS arena. Learning theories have been tested; individual differences issues have been validated against complex, real-world learning tasks (e.g., ITS, in contrast to artificial laboratory tasks); AI programming techniques have been refined; different instructional approaches have been compared; controlled studies have been conducted of aptitude-treatment interactions; and so forth. So, in terms of research vehicles, ITS are greatly underestimated. But for purposes of education, their time may be limited; maybe not.

19.7 CONCLUSIONS

Before the computer age, the prevailing instructional approach was sufficient (e.g., one teacher transmitting information to about 30 students), but we now reside in a computerized world. Initial implementations of CAI mirrored this pedantic approach, and, to some extent, so does the currently popular model-tracing approach in sophisticated ITS. Do we need to change our educational philosophies or systems?

We have most of the components necessary to advance educational reform. Not only is there great need for change, but also there are powerful, affordable technologies available to support it. Missing are definitive answers to the psychological controversies cited earlier. Basic research is actively being pursued to resolve these issues. For example, studies are beginning to find consistently that higher-order thinking skills are not acquired through didactic approaches (i.e., straight conveyance of facts) but rather through learners' active involvement with the subject matter. This "constructivist" view of learning allows students to achieve intellectual accomplishments not possible under more traditional pedagogical approaches (Collins, Brown & Newman, 1989; Resnick, 1987).

Table 19-2 contrasts old versus new approaches to instruction (from Means, Blando, Olson, Middleton, Morocco, Remz & Zorfass, 1993). The table provides a clear direction for ITS research and implementation. That is, to get from "old" to "new," we need to open up learning environments that promote increased learner initiative and between-learner collaboration. We should assess learning as it transfers to authentic tasks, not standardized tests, and attempt to establish connections across various fields so topics are not learned in isolation of one another. As technologies emerge and advance, we can fit them into this framework. Furthermore, additional research is needed to validate the goodness of the new over the old approach to teaching-learning.

Look around you. Computer technologies have dramatically transformed the workplace, communications, and commercial activities, as well as the entire business community. But education remains status quo. We need to harness the computer's potential and find ways to employ it in

TABLE 19-2. OLD VERSUS NEW APPROACHES TO INSTRUCTION

Old	New
Teacher-directed activities	Student-directed explorations
Didactic teaching	Interactive modes of instruction
Short instruction on a single subject	Extended, multidisciplinary instruction
Individual work	Collaborative work
Teacher as knowledge dispenser	Teacher as facilitator
Ability groupings	Heterogeneous groupings
Assessment of factual knowledge and discrete skills	Performance-based assessment

promoting educational change. Are current and prevalent ITS adequate for our purposes—now and in the 21st century (just right around the corner)? We believe that, as currently implemented, these systems may have asymptoted in utility. A philosophical shift has been suggested in this chapter, away from stand-alone instructional devices and toward using tools to aid in the more collaborative learning process. There are actually very few ITS in place in schools, yet they exist in abundance in research laboratories. We need to move on.

As we've discussed here, reform can proceed along a number of pathways (perhaps in parallel). For instance, computer graphics are getting better every day; we can now develop three-dimensional virtual environments where individuals can interact with any artificial world we choose to program (or purchase). Satellite transmissions can relay data to very distant locations; learners from different parts of the globe can access distal data, or even get together and jointly experience and solve various problems. Cognitive tools abound (e.g., simulators, hypertext/hypermedia formats, etc.), and we seem to be ready to recast our convictions about ITS. Rather than trying to create all-knowing, all-purpose teaching machines, a more fruitful approach may be to develop specific computerized tools. These tools can be specific for a given domain, or general purpose, applicable across domains. To paraphrase a well-known quotation: A person who is given a fish will eat for a day, but a person who learns how to fish will eat for a lifetime.

We can see the seeds of discontent growing. Go to any ITS-related conference and notice how researchers in the field have begun to discontinue using the term "ITS." Instead, in a show of semantic squirming, they refer to advanced automated instructional systems (formerly, ITS) as: interactive learning environments, cognitive tutors, individualized teaching systems, computer-assisted learning, automated instructional support systems, computer-based learning environments, immersive tutoring systems, knowledge communications systems, computer tools, and so on.

Not only is the ITS construct too ambitious, but there is no universally accepted definition of what comprises computer intelligence. While our working definition of intelligence is fairly specific, there exists a wide range of criteria in the literature related to computer-tutor intelligence. For instance, some say that for an automated instructional system to earn the label *intelligent,* it must demonstrate the ability to learn by showing an evolving knowledge base. Yazdani and Lawler (1986) asserted, "No system which is too rigid to learn should be called intelligent" (p. 201). Others have argued that intelligent systems must provide for learner control during the learning process (Papert, 1980; Scardamalia et al., 1989). Still others (e.g., MacKenzie, 1990) suggest that we reserve the word *intelligent* to describe only those systems showing truly impressive advances (e.g., intuition, empathy). Are these even realistic goals?

The fields of AI, psychology, and education have all greatly benefited from ITS research. But to continue, much more *systematic* research is needed to achieve some of the great potential offered by these systems. One suggestion is to begin a coordinated stream of systematic ITS research and development, altering specific features of existing systems and evaluating the results of those changes in accordance with a principled approach. According to Self (1989), "Once a sounder foundation for ITS has been specified, it becomes possible to identify the elements of a theory of ITS. These elements lie within (formal) AI, in areas such as belief logics, reason maintenance, metalevel architectures, and discourse models, areas from which ITS research has been divorced" (p. 244). Intelligent tutoring systems, as we now know them, may not exist 20 years from now, but we're on the right path, the motives are commendable, and the learner will ultimately profit.

As we began this review of ITS with the evolution of computer technology, so do we end it. ITS and related, developing technologies for education and training are constrained by two important factors: (a) the cost and power of computers, and (b) the pragmatic and theoretical knowledge of how best to employ them. Every month, computers are dramatically decreasing in cost and increasing in power, these changes bearing directly on consumer knowledge and application of the technology. While discussion of the interaction between these two factors goes beyond the scope of this chapter, we can make straightforward predictions about upcoming hardware and software developments. The Mips (millions of instructions per second) curve is already converging on a Bips (billions of instructions per second) curve in an exponential explosion that knows no limits. Desktop computers with 100 Mips are currently available, and this raw horsepower makes a qualitative difference in computing possibilities. Soon, powerful systems will be available in notebook- and calculator-sized formats that fit into our hands, shirt pockets, and purses. Further, software tools enable us to learn from, and perform within, all major domains, such as algebra, biology, physics, art history, computer science, home economics, psychology, botany, calculus, accounting, and even manufacturing, medicine, and engineering. With our fingertips, we will be able to retrieve information, translate foreign languages, complete our tax returns, work out investment portfolios, analyze sales trends, and so forth. Software will be everywhere with embedded "assistants" to explain, critique, provide on-line support and coaching, and perform all of the ITS activities outlined in this chapter. Society stands at the edge of all this. Although the time-line for these exciting developments is uncertain, we do know that the research conducted so far is just a drizzle in comparison with the deluge to come.

REFERENCES

Acchione-Noel, S. & Psotka, J. (1993). *MACH III: past and future approaches to intelligent tutoring.* Proceedings of the 1993 Conference on Intelligent Computer-Aided Training and Virtual Environment Technology, Houston, TX.

Alexander, P.A. & Judy, J.E. (1988). The interaction of domain-specific and strategic knowledge in academic performance. *Review of Educational Research 58*(4), 375–404.

Anderson, J.R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

— (1987). Skill acquisition: compilation of weak-method problem solutions. *Psychological Review 94,* 192–210.

— (1990). Analysis of student performance with the LISP tutor. *In* N. Fredericksen, R. Glaser, A. Lesgold & M. Shafto, eds. *Diagnostic monitoring of skill and knowledge acquisition.* Hillsdale, NJ: Erlbaum.

— (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

—, Boyle, C. & Reiser, B. (1985). Intelligent tutoring systems. *Science 228,* 456–62.

—, Boyle, C. & Yost, G. (1985). The geometry tutor. In *Proceedings of IJCAI-85,* pp. 1-7. Los Angeles, CA: IJCAI.

—, Conrad, F.G. & Corbett, A.T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science 13*(4), 467–505.

—, Farrell, R., & Sauers, R. (1984). Learning to program in LISP. *Cognitive Science 8,* 87–129.

Ausubel, D.P. (1963). *The psychology of meaningful verbal learning: an introduction to school learning.* New York: Grune & Stratton.

Badler, N.I., Phillips, C.B. & Webber, B.L. (1992). *Virtual humans and simulated agents.* New York: Oxford University Press.

Baker, E.L. (1990). Technology assessment: Policy and methodological issues. *In* H.L. Burns, J. Parlett & C. Luckhardt, eds. *Intelligent tutoring systems: evolutions in design.* Hillsdale, NJ: Erlbaum.

Barnard, Y.F., Erkens, G. & Sandberg, J.A.C. (1990). Interaction in intelligent tutoring systems. *Journal of Structural Learning 10*(3), 197–213.

Barr, A., Beard, M. & Atkinson, R.C. (1976). The computer as a tutorial laboratory: the Stanford BIP Project. *International Journal of Man-Machine Studies 8,* 567–96.

Bartlett, F.C. (1932). *Remembering: a study in experimental and social psychology.* London, England: Cambridge University Press.

Benedikt, M., ed. (1991). *Cyberspace: first steps.* Cambridge, MA: MIT Press.

Bessemer, D.W. (1991). *Transfer of SIMNET training in the army officer basic course.* (ARI Technical Report #120.) Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Bierman, D., Breuker, J. & Sandberg, J., eds. (1989). *Artificial intelligence and education: synthesis and reflection.* Springfield, VA: IOS.

Bloom, B.S. (1956). Taxonomy of educational objectives: the classification of educational goals. *In* B.S. Bloom, ed. *Cognitive domain, handbook 1.* New York: McKay.

— (1984). The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher 13*(6), 4-16.

Bonar, J., Cunningham, R., Beatty, P. & Weil, W. (1988). Bridge: intelligent tutoring system with intermediate representations (technical report). Pittsburgh, PA: University of Pittsburgh, Learning Research & Development Center.

Bower, G.H. & Hilgard, E.R. (1981). *Theories of learning.* Englewood Cliffs, NJ: Prentice Hall.

Briggs, L.J., Campeau, P.L., Gagné, R.M. & May, M.A. (1967). *Instructional media: a procedure for the design of multi-media instruction, a critical review of research and suggestions for further research.* Pittsburgh, PA: American Institutes for Research.

Brooks, R.A. (1991). Intelligence without representation. *Artificial Intelligence 47,* 139–59.

Brown, A.L. & Palincsar, A.S. (1989). Guided, cooperative learning and individual knowledge acquisition. *In* L.B. Resnick, ed. *Knowing, learning, and instruction: essays in honor of Robert Glaser,* 393–451. Hillsdale, NJ: Erlbaum.

Brown, J.S. & Burton, R.R. (1975). Multiple representations of knowledge for tutorial reasoning. *In* D.G. Bobrow & A. Collins, eds. *Representation and understanding,* 311–49. New York: Academic.

— & — (1978). *An investigation of computer coaching for informal learning activities.* Technical Report, Defense Advanced Research Projects Agency, Human Resources Lab., Lowry AFB, CO.

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

—, Burton, R.R. & deKleer, J. (1982). Pedagogical natural language, and knowledge engineering techniques in SOPHIE I, II, and III. *In* D. Sleeman & J.S. Brown, eds. *Intelligent tutoring systems,* 227–282. New York: Academic.

Bruner, J.S. (1961). The act of discovery. *Harvard Educational Review 31,* 21–32.

Bull, S., Pain, H. & Brna, P. (1993). Collaborative student modelling: cooperation between the student and system [summary]. *In* P. Brna, S. Ohlsson & H. Pain, eds. *Proceedings of AI-ED '93: World Conference on Artificial Intelligence in Education,* 547.

Bunderson, C.V. & Olsen, J.B. (1983). *Mental errors in arithmetic skills: their diagnosis in precollege students* (final project report, NSF SED 80-125000). Provo, UT: WICAT Education Institution.

Burns, H.L. & Capps, C.G. (1988). Foundations of intelligent tutoring systems: an introduction. *In* M.C. Polson & J.J. Richardson, eds. *Foundations of intelligent tutoring systems,* 1–19. Hillsdale, NJ: Erlbaum.

Burton, R.R. (1982). Diagnosing bugs in a simple procedural skill. *In* D.H. Sleeman & J.S. Brown, eds. *Intelligent tutoring systems,* 157–183. New York: Academic.

— & Brown, J.S. (1976). A tutoring and student modeling paradigm for gaming environments. *In* R. Colman & P. Lorton, Jr., eds. *Computer Science and Education. ACM SIGCSE Bulletin 8*(1), 236–46.

— & — (1982). An investigation of computer coaching for informal learning activities. *In* D. Sleeman & J.S. Brown, eds. *Intelligent tutoring systems.* London, England: Academic.

Carbonell, J.R. (1970). AI in CAI: an artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems 11*(4), 190–202.

Carr, B. & Goldstein, I.P. (1977). *Overlays: a theory of modeling for computer-aided instruction.* AI Lab Memo 406 MIT, Cambridge, MA.

Carroll, J. (1963). A model of school learning. *Teachers College Record 64,* 723–33.

Charniak, E. & McDermott, D. (1985). *Introduction to artificial intelligence.* Reading, MA: Addison-Wesley.

Clancey, W.J. (1979). Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies 11* (9), 25–49.

— (1986). *Intelligent tutoring systems: a tutorial survey.* (Report No. KSL-86-58.) Stanford, CA: Stanford University Press.

— (1992). Representation of knowing: in defense of cognitive apprenticeship. *Journal of Artificial Intelligence in Education*

3, 139–68.
— (1993). Situated action: a neuropsychological interpretation response to Vera and Simon. *Cognitive Science 17*, 87–116.
Cognition and Technology Group at Vanderbilt (1992). An anchored instruction approach to cognitive skills acquisition and intelligent tutoring. *In* J.W. Regian & V.J. Shute, eds. *Cognitive approaches to automated instruction*, 135–70, Hillsdale, NJ: Erlbaum.
Collins, A. (1977). Processes in acquiring knowledge. *In* R.C. Anderson, R.J. Spiro & W.E. Montague, eds. *Schooling and the acquisition of knowledge*, 339–63. Hillsdale, NJ: Erlbaum.
— & Brown, J.S. (1988). The computer as a tool for learning through reflection. *In* H. Mandl & A. Lesgold, eds. *Learning issues for intelligent tutoring systems*, 1–18. New York: Springer.
—, — & Newman, S.E. (1989). Cognitive apprenticeship: teaching the craft of reading, writing, and mathematics. *In* L.B. Resnick, ed. *Cognition and instruction: issues and agendas*. Hillsdale, NJ: Erlbaum.
Corbett, A.T. & Anderson, J.R. (1989). Feedback timing and student control in the LISP intelligent tutoring system. *In* D. Bierman, J. Brueker & J. Sandberg, eds. *Artificial intelligence and education: synthesis and reflection*, 64–72. Springfield, VA: IOS.
Cronbach, L.J. & Snow, R.E. (1981). *Aptitudes and instructional methods: a handbook for research on interactions*. New York: Irvington.
Crowder, N.A. (1959). Automatic tutoring by means of intrinsic programming. *In* E. Galanter, ed. *Automatic teaching: the state of the art*. New York: Wiley.
Cummings, G. & Self, J. (1989). Collaborative intelligent educating systems. *In* D. Bierman, J. Brueker & J. Sandberg, eds. *Artificial intelligence and education: synthesis and reflection*, 73–80. Springfield, VA: IOS.
Curtis, P. & Nichols, D.A. (1993). MUDs grow up: social virtual reality in the real world. *Proceedings of the 3d annual Cyberspace Conference*, Austin, TX.
Derry, S.J. & Lajoie, S.P. (1993). A middle camp for (un)intelligent instructional computing: an introduction. *In* S.P. Lajoie & S.J. Derry, eds. *Computers as cognitive tools*, 1–11. Hillsdale, NJ: Erlbaum.
Dochy, F.J.R.C. (1992). *Assessment of prior knowledge as a determinant for future learning*. Heerlen: Open University of the Netherlands.
Drescher, G.L. (1991). *Made-up minds: a constructivist approach to artificial intelligence*. Cambridge, MA: MIT Press.
Edelman, G.M. (1987). *Neural Darwinism: the theory of neuronal group selection*. New York: Basic Books.
Farr, M.J. & Psotka, J. (1992). Introduction. *In* M.J. Farr & J. Psotka, eds. *Intelligent instruction by computer: theory and practice*. Washington, DC.: Taylor & Francis.
Feurzeig, W. (1988). Apprentice tools: students as practitioners. *In* R.S. Nickerson & P.P. Zodhiates, eds. *Technology in education: looking toward 2020*. Hillsdale, NJ: Erlbaum.
Fox, B.A. (1991). Cognitive and interactional aspects of correction in tutoring. *In* P. Goodyear, ed. *Teaching knowledge and intelligent tutoring*, 149–72. Hillsdale, NJ: Ablex.
Frederiksen, J.R., White, B.Y., Collins, A. & Eggan, G. (1988). Intelligent tutoring systems for electronic troubleshooting. *In* J. Psotka, L.D. Massey & S.A. Mutter, eds. *Intelligent tutoring systems: lessons learned*, 351–68. Hillsdale, NJ: Erlbaum.
Friedman, P.G. & Yarbrough, E.A. (1985). *Training strategies from start to finish*. Englewood Cliffs, NJ: Prentice Hall.
Gagné, R.M. (1965). *The conditions of learning*. New York: Holt.
Glaser, R. (1976). The processes of intelligence and education. *In* L.B. Resnick, ed. *The nature of intelligence*, 341–52. Hillsdale, NJ: Erlbaum.
—(1984). Education and thinking: the role of knowledge. *American Psychologist 39*(2), 93–104.
Goldstein, I.P. (1979). The genetic graph: a representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies 11*, 51–77.
Goodyear, P., ed. (1991). *Teaching knowledge and intelligent tutoring*. Norwood, NJ: Ablex.
Greeno, J.G. (1989). Situations, mental models, and generative knowledge. *In* D. Klahr & K. Kotovsky, eds. *Complex information processing: the impact of Herbert A. Simon*, 285–318. Hillsdale, NJ: Erlbaum.
Gugerty, L. (1993). *Non-diagnostic intelligent tutoring systems: learning collaboratively without student models*. Manuscript submitted for publication.
Harel, I. (1991). *Children designers*. Norwood, NJ: Ablex.
— & Papert, S. (1991). *Constructionism*. Norwood, NJ: Ablex.
Hartley, J.R. & Sleeman, D.H. (1973). Towards more intelligent teaching systems. *International Journal of Man-Machine Studies 2*, 215–36.
Hollan, J.D., Hutchins, E.L. & Weitzman, L. (1984). STEAMER: an interactive inspectable simulation-based training system. *The AI Magazine 2*, 15–27.
Johnson, W.L. (1986). *Intention-based diagnosis of novice programming errors*. Research notes in artificial intelligence. Los Altos, CA: Kaufmann (copublished with Pitman, London).
— & Soloway, E.M. (1984). PROUST: knowledge-based program debugging. In *Proceedings of the Seventh International Software Engineering Conference*, 369–80, Orlando, FL.
Justen, J.E., Waldrop, T.M. & Adams, T.M. (1990). Effects of paired versus individual user computer-assisted instruction and type of feedback on student achievement. *Educational Technology 30*(7), 51–53.
Katz, S. & Lesgold, A. (1993). The role of the tutor in computer-based collaborative learning situations. *In* S.P. Lajoie & S.J. Derry, eds. *Computers as cognitive tools*, 289–318. Hillsdale, NJ: Erlbaum.
Kedar, S., Baudin, C., Birnbaum, L., Osgood, R. & Bareiss, R. (1993). Ask how it works: an interactive intelligent manual for devices. *Proceedings of INTERCHI*, 171–72, Amsterdam.
Kolodner, J.L., ed. (1988). *Proceedings of the First Case-Based Reasoning Workshop*. Los Altos, CA: Kaufmann.
Kozak, J.J., Hancock, P.A., Arthur, E.J. & Chrysler, S.T. (1993). Transfer of training from virtual reality. *Ergonomics 36*(7), 777–84.
Kurland, L.C., Granville, R.A. & MacLaughlin, D.M. (1992). Design, development, and implementation of an intelligent tutoring system for training radar mechanics to troubleshoot. *In* M.J. Farr & J. Psotka, eds. *Intelligent instruction by computer: theory and practice*, 205–238. Washington, DC: Taylor & Francis.
Kyllonen, P.C. & Christal, R.E. (1990). Cognitive modeling of learning abilities: a status report of LAMP. *In* R. Dillon & J.W. Pellegrino, eds. *Testing: theoretical and applied issues*, 112–37. San Francisco, CA: Freeman.
— & Shute, V.J. (1989). A taxonomy of learning skills. *In* P.L. Ackerman, R.J. Sternberg & R. Glaser, eds. *Learning and*

individual differences, 117-63. New York: Freeman.

Lajoie, S.P. & Derry, S.J., eds. (1993). *Computers as cognitive tools.* Hillsdale, NJ: Erlbaum.

— & Lesgold, A. (1992). Apprenticeship training in the workplace: a computer-coached practice environment as a new form of apprenticeship. *In* M. Farr & J. Psotka, eds. *Intelligent instruction by computer: theory and practice,* 15–36. New York: Taylor & Francis.

Lampert, M. (1986). Knowing, doing, and teaching multiplication. *Cognition and Instruction 3*(4), 305–42.

Lave, J. & Wenger, E. (1991). *Situated learning: legitimate peripheral participation.* Cambridge, MA: Cambridge University Press.

Lawler, R.W., & Yazdani, M. (1987). *Artificial intelligence and education.* Norwood, NJ: Ablex.

Lepper, M.R., Aspinwall, L., Mumme, D. & Chabay, R.W. (1990). Self-perception and social perception processes in tutoring: subtle social control strategies of expert tutors. *In* J.M. Olson & M.P. Zanna, eds. *Self inference processes: the sixth Ontario symposium in social psychology,* 217–37. Hillsdale, NJ: Erlbaum.

Lesgold, A. (1988). Toward a theory of curriculum for use in designing intelligent instructional systems. *In* H. Mandl & A. Lesgold, eds. *Learning issues for intelligent tutoring systems,* 114–37. New York: Springer.

—, Eggan, G., Katz, S. & Rao, G. (1992). Possibilities for assessment using computer-based apprenticeship environments. *In* J.W. Regian & V.J. Shute, eds. *Cognitive approaches to automated instruction,* 49–80. Hillsdale, NJ: Erlbaum.

—, Lajoie, S.P., Bunzo, M. & Eggan, G. (1992). A coached practice environment for an electronics troubleshooting job. *In* J. Larkin, R. Chabay & C. Sheftic, eds. *Computer-assisted instruction and intelligent tutoring systems: establishing communication and collaboration.* Hillsdale, NJ: Erlbaum.

Lewis, M.W., McArthur, D., Stasz, C. & Zmuidzinas, M. (1990). Discovery-based tutoring in mathematics. *AAAI Spring Symposium Series.* Stanford University, Stanford, CA.

Littman, D. & Soloway, E. (1988). Evaluating ITSs: the cognitive science perspective. *In* M.C. Polson & J.J. Richardson, eds. *Foundations of intelligent tutoring systems,* 209–42. Hillsdale, NJ: Erlbaum.

Loftin, B. & Dede, C. (1993). Described in surreal science. *Scientific American,* Feb., p. 103.

MacKenzie, I.S. (1990). Courseware evaluation: where's the intelligence? *Journal of Computer Assisted Learning 6,* 273–85.

Means, B., Blando, J., Olson, K., Middleton, T., Morocco, C.C., Remz, A.R. & Zorfass, J. (1993). *Using technology to support education reform.* Washington, DC: U.S. Government Printing Office.

Merrill, D.C., Reiser, B.J. & Landes, S. (1992). *Human tutoring: pedagogical strategies and learning outcomes.* Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA.

—, Reiser, B.J., Ranney, M. & Trafton, G.J. (1992). Effective tutoring techniques: a comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences.*

Miller, M.L. & Lucado, S.R. (1992). Integrating intelligent tutoring, computer-based training, and interactive video in a prototype maintenance trainer. *In* M.J. Farr & J. Psotka, eds. *Intelligent instruction by computer: theory and practice,* 127–50. Washington, DC: Taylor & Francis.

Minstrell, J. (1988). Teachers' assistants: what could technology make feasible? *In* R.S. Nickerson & P.P. Zodhiates, eds. *Technology in education: looking toward 2020.* Hillsdale, NJ: Erlbaum.

Neisser, U. (1987). A sense of where you are: functions of the spatial module. *In* P. Ellen & C. Thinus-Blanc, eds. *Cognitive processes and spatial orientation in animal and man, Vol. II: neurophysiology and developmental aspects,* 293–310. Boston, MA: Martinus Nijhoff.

Nichols, P., Pokorny, R., Jones, G., Gott, S.P. & Alley, W.E. (1995). *Evaluation of an avionics troubleshooting tutoring system.* Technical Report, Armstrong Laboratory, Human Resources Directorate, Brooks AFB, TX.

Nickerson, R.S. (1988). Technology in education: possible influences on context, purposes, content, and methods. *In* R.S. Nickerson & P.P. Zodhiates, eds. *Technology in education: looking toward 2020.* Hillsdale, NJ: Erlbaum.

— & Zodhiates, P.P., eds. (1988). *Technology in education: looking toward 2020.* Hillsdale, NJ: Erlbaum.

Ohlsson, S. (1986). Some principles of intelligent tutoring. *Instructional Science 14,* 293–326.

Otsuki, S. (1993). Intelligent environment for discovery learning. *In* P. Brna, S. Ehlsson & H. Pain, eds. *Proceedings from Artificial Intelligence in Education,* 15–20. Charlottesville, VA: AACE.

Palincsar, A.S. & Brown, A.L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction 1,* 117–75.

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

Piaget, J. (1954). *The construction of reality in the child.* New York: Ballentine.

Polson, M.C. & Richardson, J.J., eds. (1988). *Foundations of intelligent tutoring systems.* Hillsdale, NJ: Erlbaum.

Pressey, S.L. (1926). A simple apparatus which gives tests and scores-and-teaches. *School and Society 23,* 373–76.

Psotka, J. (1993, May). *Virtual egocenters as a function of display geometric field of view and eye station point.* Proceedings of 2d Conference on Intelligent Computer Aided Instruction and Synthetic Environments, Houston, TX.

—, Holland, M. & Kerst, S. (1992). The technological promise for second language intelligent tutoring systems in the 21st century. *In* M. Swartz & M. Yazdani, eds. *Intelligent tutoring systems for foreign language learning: the bridge to international communication.* Berlin: Springer.

—, Massey, L.D. & Mutter, S.A. (1988). *Intelligent tutoring systems: lessons learned.* Hillsdale, NJ: Erlbaum.

Regian, J.W. & Shute, V.J., eds. (1992). *Cognitive approaches to automated instruction.* Hillsdale, NJ: Erlbaum.

—, Shebilske, W. & Monk, J. (1992). A preliminary empirical evaluation of virtual reality as an instructional medium for visual-spatial tasks. *Journal of Communication 42* (4), 136–49.

Regian, J.W., Shebilske, W. & Monk, J. (1993). *VR as a training tool: transfer effects.* Unpublished manuscript. Armstrong Laboratory, Brooks Air Force Base, TX.

Reiser, B.J., Ranney, M., Lovett, M.C. & Kimberg, D.Y. (1989). Facilitating students' reasoning with causal explanations and visual representations. *In* D. Bierman, J. Breuker & J. Sandberg, eds. *Artificial intelligence and education: synthesis and reflection,* 228–35. Springfield, VA: IOS.

Resnick, L.B. (1987). *Education and learning to think.* Washington, DC: National Academy.

Resnick, L.B. & Johnson, A. (1988). Intelligent machines for intelligent people: cognitive theory and the future of computer-assisted learning. *In* R.S. Nickerson & P.P. Zodhiates, eds. *Technology in education: looking toward 2020.* Hillsdale, NJ: Erlbaum.

Riesbeck, C.K. & Schank, R.C. (1990). From training to teaching: techniques for case-based ITS. *In* H. Burns, C. Luckhardt & J. Parlett, eds. *Knowledge architectures in intelligent tutoring systems.* Orlando, FL: Academic.

Salomon, G. (1993). On the nature of pedagogic computer tools: the case of the *writing partner. In* S.P. Lajoie & S.J. Derry, eds. *Computers as cognitive tools,* 179–96. Hillsdale, NJ: Erlbaum.

Scardamalia, M., Bereiter, C., McLean, R.S., Swallow, J. & Woodruff, E. (1989). Computer-supported intentional learning environments. *Journal of Educational Computing Research 5*(1), 51–68.

Schank, R.C. (1982). *Dynamic memory: a theory of learning in computers and people.* Cambridge, MA: Cambridge University Press.

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

Schofield, J.W. & Evans-Rhodes, D. (1989) Artificial intelligence in the classroom. *In* D. Bierman, J. Breuker & J. Sandberg, eds. *Artificial intelligence and education: synthesis and reflection,* 238–43. Springfield, VA: IOS.

Self, J.A., ed. (1988). *Artificial intelligence and human learning: intelligent computer-aided instruction.* London, England: Chapman & Hall.

— (1989). The case for formalising student models (and intelligent tutoring systems generally). *In* D. Bierman, J. Breuker & J. Sandberg, eds. *Artificial intelligence and education: synthesis and reflection,* 244. Springfield, VA: IOS.

Shortliffe, E.H. (1976). *Computer-based medical consultations: MYCIN.* Amsterdam, Holland: Elsevier.

Shute, V.J. (1991). Who is likely to acquire programming skills? *Journal of Educational Computing Research 7,* 1–24.

— (1992). Aptitude-treatment interactions and cognitive skill diagnosis. *In* J.W. Regian & V.J. Shute, eds. *Cognitive approaches to automated instruction,* 15–47. Hillsdale, NJ: Erlbaum.

— (1993-a). A macroadaptive approach to tutoring. *Journal of Artificial Intelligence and Education 4*(1), 61–93.

— (1993b). A comparison of learning environments: all that glitters . . . *In* S.P. Lajoie & S.J. Derry, eds. *Computers as cognitive tools,* 47–74. Hillsdale, NJ: Erlbaum.

— (1994, Apr.). *Discovery learning environments: appropriate for all?* Paper presented at the American Educational Research Association, New Orleans, LA.

— (1995). SMART: Student modeling approach for responsive tutoring. *User Modeling and User-Adapted Interaction 5,* 1–44.

— , Lajoie, S.P. & Gluck, K.A. (in press). Individualized and group approaches to training. To appear in S. Tobias & D. Fletcher, eds., *Handbook on training.*

— & Gawlick-Grendell, L.A. (1994). *What does the computer contribute to learning?* Manuscript submitted for publication.

—, — & Young, R. (1993, Apr.). *An experiential system for learning probability: Stat Lady.* Paper presented at the American Educational Research Association, Atlanta, GA.

— & Glaser, R. (1990). A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments (1),* 51–76.

— & — (1991). An intelligent tutoring system for exploring principles of economics. *In* R.E. Snow & D. Wiley, eds. *Improving inquiry in social science: a volume in honor of Lee J. Cronbach,* 333–66. Hillsdale, NJ: Erlbaum.

—, — & Raghavan, K. (1989). Inference and discovery in an exploratory laboratory. *In* P.L. Ackerman, R.J. Sternberg & R. Glaser, eds. *Learning and individual differences,* 279–326. New York: Freeman.

—, Lajoie, S.P. & Gluck, K.A. (in press). Individualized and group approaches to training. To appear in S. Tobias & D. Fletcher, eds. *Handbook on training.*

— & Regian, J.W. (1990). Rose garden promises of intelligent tutoring systems: blossom or thorn? *Proceedings from the Space Operations, Applications and Research Symposium,* Albuquerque, NM.

— & — (1993). Principles for evaluating intelligent tutoring systems. In a special evaluation issue of *Journal of Artificial Intelligence & Education 4*(3), 245–71.

Skinner, B.F. (1957). *Verbal behavior.* Englewood Cliffs, NJ: Prentice Hall.

Sleeman, D.H. (1984). *Intelligent tutoring systems: a review* (Report No. IR011683). Stanford, CA: Stanford University, School of Education & Department of Computer Science. (ERIC Document Reproduction Service No. ED 257 450.)

— (1987). PIXIE: a shell for developing intelligent tutoring systems. *In* R. Lawler & M. Yazdani, eds. *AI and education: learning environments and intelligent tutoring systems,* 239–65. Norwood, NJ: Ablex .

— & Brown, J.S. (1982). *Intelligent tutoring systems.* London, England: Academic.

—, Kelly, A.E., Martinak, R., Ward, R.D. & Moore, J.L. (1989). Studies of diagnosis and remediation with high school algebra students. *Cognitive Science 13*(4), 551–68.

Spencer, R.W. & Van Eynde, D.F. (1986, Fall). Experiential learning in economics. *Journal of Economic Education,* 289–94.

Stansfield, J.C., Carr, B. & Goldstein, I.P. (1976). Wumpus advisor I: a first implementation of a program that tutors logical and probabilistic reasoning skills. *AI Lab Memo 381.* MIT, Cambridge, MA.

Stevens, A.L. & Collins, A. (1977). The goal structure of a Socratic tutor. In *Proceedings of the National ACM Conference,* Seattle, WA, 256–63. New York: Association for Computing Machinery.

Suchman, L.A. (1987). *Plans and situated actions.* Cambridge, MA: Cambridge University Press.

Swan, M.B. (1983). *Teaching decimal place value: a comparative study of conflict and positively-only approaches* (Research Rep. No. 31). Nottingham, England: University of Nottingham, Sheel Center for Mathematical Education.

Sweller, J. (1988). Cognitive load during problem solving: effects on learning. *Cognitive Science 12,* 257–85.

Teasley, S.D. & Roschelle, J. (1993). Constructing a joint problem space: the computer as a tool for sharing knowledge. *In* S.P. Lajoie & S.J. Derry, eds. *Computers as cognitive tools,* 229–58. Hillsdale, NJ: Erlbaum.

Thorisson, K.R. (1993). Dialogue control in social interface agents. *Proceedings of INTERCHI,* 139–40. Amsterdam, Holland.

Towne, D.M. & Munro, A. (1992). Two approaches to simulation composition for training. *In* M.J. Farr & J. Psotka, eds.

Intelligent instruction by computer: theory and practice, 105–25. Washington, DC: Taylor & Francis.

Turkle, S. (1993, May). Constructions and reconstructions of the self in virtual reality. *Proceedings of the 3d annual Cyberspace Conference,* Austin, TX.

Uhr, L. (1969). Teaching machine programs that generate problems as a function of interaction with students. *Proceedings of the 24th National Conferences,* 125–34.

VanLehn, K. (1990). *Mind bugs: the origins of procedural misconceptions.* Cambridge, MA: MIT Press.

Vera, A.H., & Simon, H.A. (1993). Situated action: reply to William Clancey. *Cognitive Science 17*(1), 117–33.

Wenger, E. (1987). *Artificial intelligence and tutoring systems.* Los Altos, CA: Kaufmann.

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

— & Frederiksen, J.R. (1987). Qualitative models and i gent learning environments. *In* R. Lawler & M. Ya eds. *AI and education,* 281–305. Norwood, NJ: Ablex

— & Horowitz, P. (1987). *Thinker tools: enabling child understand physical laws* (Report No. 6470). Camb MA: Bolt, Beranek & Newman.

Woolf, B.P. (1988). Intelligent tutoring systems: a surv H. Schrobe, ed. *Exploring artificial intelligence,* 1–44. Alto, CA: Kaufmann.

Yazdani, M. & Lawler, R.W. (1986). Artificial intelligenc education: an overview. *Instructional Science 14,* 197

Yob, G. (1975, Sep./Oct.). Hunt the Wumpus. *Creative puting,* 51–54.